

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 4 年 1 月 9 日
Date of Application:

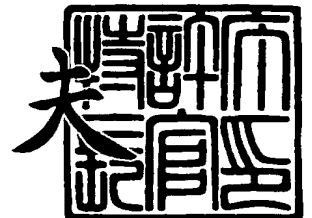
出 願 番 号 特 願 2 0 0 4 - 0 0 4 1 2 3
Application Number:
[ST. 10/C]: [J P 2 0 0 4 - 0 0 4 1 2 3]

出 願 人 株式会社リコー
Applicant(s):

2 0 0 4 年 1 月 2 6 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 0309908
【提出日】 平成16年 1月 9日
【あて先】 特許庁長官 今井 康夫 殿
【国際特許分類】 G03G 15/00
G06F 17/21
【発明者】
【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内
【氏名】 松田 透
【特許出願人】
【識別番号】 000006747
【氏名又は名称】 株式会社リコー
【代理人】
【識別番号】 100070150
【弁理士】
【氏名又は名称】 伊東 忠彦
【先の出願に基づく優先権主張】
【出願番号】 特願2003- 16427
【出願日】 平成15年 1月24日
【手数料の表示】
【予納台帳番号】 002989
【納付金額】 21,000円
【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9911477

【書類名】 特許請求の範囲**【請求項 1】**

一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理装置において、

前記複数の符号化方式を選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を前記複数の符号化方式ごとに取得して、前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定する符号化方式決定手段を有することを特徴とする文字列処理装置。

【請求項 2】

前記符号化方式決定手段は、前記複数の符号化方式を順番に選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方が所定基準を満たさないときに、前記複数の符号化方式から次の符号化方式を選択することを特徴とする請求項 1 記載の文字列処理装置。

【請求項 3】

前記符号化方式決定手段は、前記複数の符号化方式を優先度の順番に選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方が所定基準を満たさないときに、前記複数の符号化方式から次の符号化方式を選択することを特徴とする請求項 1 記載の文字列処理装置。

【請求項 4】

前記複数の符号化方式の優先度は、オペレーションパネルの表示言語として設定されている言語に応じて変化することを特徴とする請求項 3 記載の文字列処理装置。

【請求項 5】

前記複数の符号化方式の優先度は、文字列の変換を指示したクライアントの表示言語として設定されている言語に応じて変化することを特徴とする請求項 3 記載の文字列処理装置。

【請求項 6】

前記符号化方式決定手段は、前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定できないときに、前記複数の符号化方式の優先度に基づき前記他の符号化方式を決定することを特徴とする請求項 1 記載の文字列処理装置。

【請求項 7】

前記符号化方式決定手段は、前記選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方が所定基準を満たすときに、前記選択した符号化方式を前記他の符号化方式として決定することを特徴とする請求項 1 記載の文字列処理装置。

【請求項 8】

前記代用符号は、前記一の符号化方式で符号化された文字列のうち、前記選択した符号化方式を用いて変換できない文字を置き換えた文字コードであることを特徴とする請求項 1 乃至 7 何れか一項記載の文字列処理装置。

【請求項 9】

一の符号化方式で符号化された文字列を用いて、他の符号化方式で符号化された文字列を照合する文字列処理装置において、

前記他の符号化方式で符号化された文字列を前記一の符号化方式に変換し、変換した文字列を前記一の符号化方式で符号化された文字列を用いて照合するときに、前記変換した文字列に含まれる代用符号を、全ての文字を表す役割をもった文字として扱う文字列照合手段

を有することを特徴とする文字列処理装置。

【請求項 10】

前記他の符号化方式で符号化される前の文字列を格納している格納手段を更に有し、前

記他の符号化方式で符号化された文字列を前記一の符号化方式で符号化された文字列を用いて照合するときに、前記格納手段に格納している前記他の符号化方式で符号化される前の文字列を用いることを特徴とする請求項 9 記載の文字列処理装置。

【請求項 1 1】

前記一の符号化方式で符号化された文字のうち、同一の文字に異なる文字コードを割り当てている例を設定したテーブルを更に有することを特徴とする請求項 9 又は 1 0 記載の文字列処理装置。

【請求項 1 2】

前記代用符号は、前記他の符号化方式で符号化された文字列のうち、前記一の符号化方式に変換できない文字を置き換えた文字コードであることを特徴とする請求項 9 乃至 1 1 何れか一項記載の文字列処理装置。

【請求項 1 3】

一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理方法において、

前記複数の符号化方式を選択する段階と、

選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を前記複数の符号化方式ごとに取得する段階と、

前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定する段階と

を有することを特徴とする文字列処理方法。

【請求項 1 4】

一の符号化方式で符号化された文字列を用いて、他の符号化方式で符号化された文字列を照合する文字列処理方法において、

前記他の符号化方式で符号化された文字列を前記一の符号化方式に変換する段階と、

前記変換した文字列に含まれる代用符号を全ての文字を表す役割をもった文字として扱って、前記変換した文字列を前記一の符号化方式で符号化された文字列を用いて照合する段階と

を有することを特徴とする文字列処理方法。

【請求項 1 5】

前記代用符号は、前記一の符号化方式または他の符号化方式に変換できない文字を置き換えた文字コードであることを特徴とする請求項 1 3 又は 1 4 記載の文字列処理方法。

【請求項 1 6】

一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理機能を有した画像形成装置であって、

前記複数の符号化方式を選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を前記複数の符号化方式ごとに取得して、前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定する符号化方式決定手段

を有することを特徴とする画像形成装置。

【請求項 1 7】

一の符号化方式で符号化された文字列を用いて、他の符号化方式で符号化された文字列を照合する文字列照合機能を有した画像形成装置であって、

前記他の符号化方式で符号化された文字列を前記一の符号化方式に変換し、変換した文字列を前記一の符号化方式で符号化された文字列を用いて照合するときに、前記変換した文字列に含まれる代用符号を、全ての文字を表す役割をもった文字として扱う文字列照合手段

を有することを特徴とする画像形成装置。

【請求項 1 8】

前記代用符号は、前記一の符号化方式または他の符号化方式に変換できない文字を置き

● 換えた文字コードであることを特徴とする請求項 1 6 又は 1 7 記載の画像形成装置。

【書類名】 明細書

【発明の名称】 文字列処理装置、文字列処理方法および画像形成装置

【技術分野】

【0001】

本発明は、文字列処理装置、文字列処理方法および画像形成装置に係り、特に一の符号化方式（文字コード・セット）で符号化された文字列を他の符号化方式に変換する文字列処理装置、文字列処理方法および画像形成装置に関する。

【背景技術】

【0002】

符号化方式は、文字や記号など（以下、単に文字という）をコンピュータ上で扱うため、文字に割り当てられている文字コードに変換するものである。この符号化方式は、言語やコンピュータシステムにより異なるのが普通である。例えばインターネットでは、世界の主要言語に対応するためUnicodeを利用したUTF-8、UTF-16などが標準的な符号化方式として利用されている。また、文字列処理装置および画像形成装置などでは、シフトJISやLatin1等が符号化方式として利用されている。

【0003】

複数の符号化方式を利用可能な文字列処理装置は、必要に応じて一の符号化方式で符号化された文字列を他の符号化方式に変換する。なお、画像形成装置はフォントROMの容量を節約するため、ユーザインターフェースに必要十分な少数の符号化方式を販売先の言語に応じて用意している。

【0004】

従来、インターネットなどのネットワークに接続した文字列処理装置および画像形成装置は、ネットワーク側から例えばUnicodeで表現された文字列を含む要求（例えば、文書名の変更要求など）を受信すると、その文字列を内部処理で利用する符号化方式に変換していた。特許文献1には、一の符号化方式で符号化された文字列を他の符号化方式に変換する内容が記載されている。

【特許文献1】 特表平11-512543号

【発明の開示】

【発明が解決しようとする課題】

【0005】

通常、文字列処理装置および画像形成装置は複数の符号化方式が利用可能であるため、利用する符号化方式を選択する必要がある。しかしながら、符号化方式で表現可能な文字の集合は符号化方式ごとに異なっており、変換前の符号化方式で表現可能な文字が変換後の符号化方式で表現可能な文字であるとは限らない。したがって、変換前の符号化方式と変換後の符号化方式との組み合わせによっては、全ての文字列を完全に変換できない場合があるという問題があった。

【0006】

例えばUnicodeで表現可能な文字の集合は、文字列処理装置および画像形成装置が内部処理で利用するシフトJISやLatin1等の符号化方式で表現可能な文字の集合に全て包含されているとは限らない。したがって、Unicodeで表現可能な文字であっても、内部処理で利用する符号化方式で変換可能であるとは限らないという問題があった。

【0007】

以上のように、従来の文字列処理装置および画像形成装置は、一の符号化方式で符号化された文字列をどの符号化方式に変換するかで、変換できない文字の数が異なるため、符号化方式の選択が重要であった。また、従来の文字列処理装置および画像形成装置は、変換後の文字列を照合するとき等に、変換できなかった文字の扱いが重要であった。さらに、いくつかの符号化方式では、同一の文字に異なる文字コードを割り当てており、このような例外的な文字の扱いも重要であった。

【0008】

本発明は、上記の点に鑑みなされたもので、一の符号化方式で符号化された文字列を他の符号化方式に変換するときに、適切な符号化方式を選択することができ、変換後の符号化方式で表現された文字列を適切に扱うことが可能な文字列処理装置、文字列処理方法および画像形成装置を提供することを目的とする。

【課題を解決するための手段】

【0009】

そこで、上記課題を解決するため、本発明は、一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理装置において、前記複数の符号化方式を選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を前記複数の符号化方式ごとに取得して、前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定する符号化方式決定手段を有することを特徴とする。

【0010】

また、本発明は、一の符号化方式で符号化された文字列を用いて、他の符号化方式で符号化された文字列を照合する文字列処理装置において、前記他の符号化方式で符号化された文字列を前記一の符号化方式に変換し、変換した文字列を前記一の符号化方式で符号化された文字列を用いて照合するときに、前記変換した文字列に含まれる代用符号を、全ての文字を表す役割をもった文字として扱う文字列照合手段を有することを特徴とする。

【0011】

また、本発明は、一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理方法において、前記複数の符号化方式を選択する段階と、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を前記複数の符号化方式ごとに取得する段階と、前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定する段階とを有することを特徴とする。

【0012】

また、本発明は、一の符号化方式で符号化された文字列を用いて、他の符号化方式で符号化された文字列を照合する文字列処理方法において、前記他の符号化方式で符号化された文字列を前記一の符号化方式に変換する段階と、前記変換した文字列に含まれる代用符号を全ての文字を表す役割をもった文字として扱って、前記変換した文字列を前記一の符号化方式で符号化された文字列を用いて照合する段階とを有することを特徴とする。

【0013】

また、本発明は、一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理機能を有した画像形成装置であって、前記複数の符号化方式を選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を前記複数の符号化方式ごとに取得して、前記数情報および位置情報の少なくとも一方に基づき前記他の符号化方式を決定する符号化方式決定手段を有することを特徴とする。

【0014】

また、本発明は、一の符号化方式で符号化された文字列を用いて、他の符号化方式で符号化された文字列を照合する文字列照合機能を有した画像形成装置であって、前記他の符号化方式で符号化された文字列を前記一の符号化方式に変換し、変換した文字列を前記一の符号化方式で符号化された文字列を用いて照合するときに、前記変換した文字列に含まれる代用符号を、全ての文字を表す役割をもった文字として扱う文字列照合手段を有することを特徴とする。

【0015】

本発明では、一の符号化方式で符号化された文字列を他の符号化方式の文字列に変換するときに、代用符号の数情報および位置情報の少なくとも一方に基づいて符号化方式を決定する。また、本発明では他の符号化方式に変換した文字列に含まれる代用符号を、全ての文字を表す役割をもった文字（ワイルドカード）として扱う。

【発明の効果】**【0016】**

上述の如く、本発明によれば、一の符号化方式で符号化された文字列を他の符号化方式に変換するときに、適切な符号化方式を選択することができ、変換後の符号化方式で表現された文字列を適切に扱うことが可能な文字列処理装置、文字列処理方法および画像形成装置を提供することができる。

【発明を実施するための最良の形態】**【0017】**

次に、本発明を実施するための最良の形態を、以下の実施例に基づき図面を参照しつつ説明していく。

【0018】

まず、本発明の理解を容易とするために、本発明の原理について図1～7を参照しつつ説明する。図1～6は、符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。また、図7は文字に割り当てられている文字コードを符号化方式ごとに表したルックアップテーブルである。

【0019】

図1～図6では、 Ω が全ての文字の集合を表し、Aが符号化方式aで符号化可能な文字の集合を表し、Bが符号化方式bで符号化可能な文字の集合を表し、Cが符号化方式bで符号化可能な文字の集合を表している。

【0020】

図1は、符号化方式b又はcで符号化可能な文字の集合 $B \cup C$ が、符号化方式aで符号化可能な文字の集合Aに包含されていることを表している。例えば符号化方式aがUTF-8、符号化方式bがShift_JIS、符号化方式cがLatin1であるのときに、図1のような関係を有する。なお、符号化方式がUTF-8、Shift_JIS、Latin1のときのルックアップテーブルは図7のように表される。

【0021】

図2は、符号化方式a、b及びcで符号化可能な文字の集合 $A \cap B \cap C$ が、斜線で表されている。例えば図7のルックアップテーブルでは、文字「1」が図2の斜線部分に含まれる。図2の斜線部分に含まれる文字は、一の符号化方式から他の符号化方式に変換できる。

【0022】

図3は、符号化方式a、bで符号化可能かつ符号化方式cで符号化できない文字の集合 $A \cap B - C$ が、斜線で表されている。例えば図7のルックアップテーブルでは、文字「円」が図3の斜線部分に含まれる。図3の斜線部分に含まれる文字は、一の符号化方式から符号化方式cに変換できない。

【0023】

本発明では、図3の斜線部分に含まれている文字を符号化方式cに変換する場合、その文字を代用符号に置き換える。代用符号には、文字が割り当てられていない文字コードを用いる。

【0024】

例えば代用符号は、文字を1バイトで表現可能な符号化方式を2バイトに拡張した文字コードを用いることが考えられる。また、代用符号は文字が割り当てられていない隙間の文字コードを用いることが考えられる。さらに、文字列処理装置や画像形成装置などで用いない例えば「字削除」の制御文字を用いることも考えられる。

【0025】

図4は、符号化方式a、cで符号化可能かつ符号化方式bで符号化できない文字の集合 $A \cap C - B$ が、斜線で表されている。例えば図7のルックアップテーブルでは、文字

【0026】

[外1]



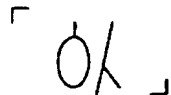
が図4の斜線部分に含まれる。図4の斜線部分に含まれる文字は、一の符号化方式から符号化方式bに変換できない。本発明では、図4の斜線部分に含まれている文字を符号化方式bに変換する場合、その文字を代用符号に置き換える。

【0027】

図5は、符号化方式aで符号化可能かつ符号化方式b, cで符号化できない文字の集合 $A - (B \cup C)$ が、斜線で表されている。例えば図7のルックアップテーブルでは、文字

【0028】

[外2]



が図5の斜線部分に含まれる。図5の斜線部分に含まれる文字は、一の符号化方式から符号化方式b, cに変換できない。本発明では、図5の斜線部分に含まれている文字を符号化方式b, cに変換する場合、その文字を代用符号に置き換える。

【0029】

図6は、符号化方式a, b, cで符号化できない文字の集合

【0030】

[外3]



が、斜線で表されている。図6の斜線部分に含まれる文字は、一の符号化方式から符号化方式a, b, cに変換できない。もっとも、符号化方式aに変換できない文字が文字コードで表現されているなら、最初から代用符号になっているものと考えられる。したがって、本発明では図6の斜線部分に含まれている文字を符号化方式b, cに変換する場合、符号化方式aの代用符号を符号化方式b, cの代用符号に置き換える。代用符号に置き換えられた文字は、例えばオペレーションパネルなどの操作画面に不明文字として「□」で表示される。

【0031】

本発明の文字列処理装置は、複数の符号化方式が利用可能であるとき、代用符号に置き換えた文字の数、代用符号に置き換えた文字の位置および符号化方式の優先度により、後述するように符号化方式を選択する。

【0032】

以下、本発明の文字列処理方法を利用する文字列処理装置の一例として画像形成装置の

処理を中心に説明するが、一の符号化方式で符号化された文字列を他の符号化方式に変換する如何なる装置であってもよい。

【0033】

なお、本実施例で説明する画像形成装置は、プリンタ、コピー、ファクシミリおよびスキャナなどの各装置の機能を1つの筐体内に収納しているため、融合機とも呼ばれる。

【0034】

この融合機は、1つの筐体内に表示部、印刷部および撮像部などを設けると共に、プリンタ、コピー、ファクシミリおよびスキャナにそれぞれ対応する4種類のソフトウェアを設け、そのソフトウェアを切り替えることより、プリンタ、コピー、ファクシミリおよびスキャナとして動作させるものである。

【0035】

図8は、本発明による融合機の一実施例の構成図である。融合機1は、ソフトウェア群2と、融合機起動部3と、ハードウェア資源4とを含むように構成される。

【0036】

融合機起動部3は融合機1の電源投入時に最初に実行され、アプリケーション層5およびプラットフォーム6を起動する。例えば融合機起動部3は、アプリケーション層5およびプラットフォーム6のプログラムをハードディスク装置（以下、HDDという）などから読み出し、読み出した各プログラムをメモリ領域に転送して起動する。

【0037】

ハードウェア資源4は、白黒レーザプリンタ（B&W LP）11と、カラーレーザプリンタ（Color LP）12と、スキャナやファクシミリなどのその他のハードウェアリソース13とを含む。

【0038】

また、ソフトウェア群2は、UNIX（登録商標）などのオペレーティングシステム（以下、OSという）上に起動されているアプリケーション層5とプラットフォーム6とを含む。アプリケーション層5は、プリンタ、コピー、ファックスおよびスキャナなどの画像形成にかかるユーザサービスにそれぞれ固有の処理を行うプログラムを含む。

【0039】

アプリケーション層5は、プリンタアプリ21と、コピーアプリ22と、ファックスアプリ23と、スキャナアプリ24と、WEBサービスインターフェース（以下、WEBサービスI/Fという）26とを含む。

【0040】

プラットフォーム6は、アプリケーション層5からの処理要求を解釈してハードウェア資源4の獲得要求を発生するコントロールサービス層9と、1つ以上のハードウェア資源4の管理を行ってコントロールサービス層9からの獲得要求を調停するシステムリソースマネージャ（以下、SRMという）39と、SRM39からの獲得要求に応じてハードウェア資源4の管理を行うハンドラ層10とを含む。

【0041】

コントロールサービス層9は、ネットワークコントロールサービス（以下、NCSという）31、デリバリーコントロールサービス（以下、DCSという）32、オペレーションパネルコントロールサービス（以下、OCSという）33、ファックスコントロールサービス（以下、FCSという）34、エンジンコントロールサービス（以下、ECSという）35、メモリコントロールサービス（以下、MCSという）36、ユーザ情報コントロールサービス（以下、UCSという）37、システムコントロールサービス（以下、SCSという）38など、一つ以上のサービスモジュールを含むように構成されている。

【0042】

なお、プラットフォーム6は予め定義されている関数により、アプリケーション層5からの処理要求を受信するアプリケーションプログラムインターフェース（以下、APIという）51を含むように構成されている。OSは、アプリケーション層5およびプラットフォーム6の各ソフトウェアをプロセスとして並列実行する。

【0043】

NCS31のプロセスは、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、各アプリケーションからのデータをネットワーク側に送信する際の仲介を行う。

【0044】

例えばNCS31のプロセスは、ネットワークを介して接続されるクライアントとのデータ通信を例えばhttpd (HyperText Transfer Protocol Daemon) により、HTTP (HyperText Transfer Protocol) で制御する。

【0045】

DCS32のプロセスは、蓄積文書の配送などの制御を行う。OCS33のプロセスは、オペレーションパネルの制御を行う。FCS34のプロセスは、アプリケーション層5からPSTN、ISDN網を利用したファックス送受信、バックアップ用のメモリで管理されている各種ファックスデータの登録／引用、ファックス読み取り、ファックス受信印刷などを行うためのAPIを提供する。

【0046】

ECS35のプロセスは、白黒レーザプリンタ11、カラーレーザプリンタ12、ハードウェアリソース13などのエンジン部の制御を行う。MCS36のプロセスは、メモリの取得および解放、HDDの利用、画像データの圧縮および伸張などのメモリ制御を行う。UCS37のプロセスは、ユーザ情報の管理を行うものである。

【0047】

SCS38のプロセスは、アプリケーション管理、操作部制御、システム画面表示、LED表示、ハードウェア資源管理、割り込みアプリケーション制御などの処理を行う。

【0048】

SRM39のプロセスは、SCS38と共にシステムの制御およびハードウェア資源4の管理を行うものである。例えばSRM39のプロセスは、白黒レーザプリンタ11やカラーレーザプリンタ12などのハードウェア資源4を利用する上位層からの獲得要求に従って調停を行い、実行制御する。

【0049】

具体的に、SRM39のプロセスは獲得要求されたハードウェア資源4が利用可能であるか（他の獲得要求により利用されていないかどうか）を判定し、利用可能であれば獲得要求されたハードウェア資源4が利用可能である旨を上位層に通知する。また、SRM39のプロセスは上位層からの獲得要求に対してハードウェア資源4を利用するためのスケジューリングを行い、要求内容（例えば、プリンタエンジンによる紙搬送と作像動作、メモリ確保、ファイル生成など）を直接実施している。

【0050】

また、ハンドラ層10は後述するファックスコントロールユニット（以下、FCUという）の管理を行うファックスコントロールユニットハンドラ（以下、FCUHという）40と、プロセスに対するメモリの割り振り及びプロセスに割り振ったメモリの管理を行うイメージメモリハンドラ（以下、IMHという）41とを含む。

【0051】

SRM39、FCUH40およびIMH41は、予め定義されている関数によりハードウェア資源4に対する処理要求を送信するエンジンI/F52を利用してハードウェア資源4に対する処理要求を行う。

【0052】

図8のような構成により、融合機1は各アプリケーションで共通的に必要な処理をプラットフォーム6で一元的に処理することができる。

【0053】

次に、融合機1のハードウェア構成について説明する。図9は、本発明による融合機の一実施例のハードウェア構成図である。融合機1は、コントローラ60と、オペレーションパネル70と、FCU80と、USBデバイス90と、IEEE1394デバイス10

0 と、エンジン部 110 とを含む。

【0054】

また、コントローラ 60 は、CPU 61 と、システムメモリ 62 と、ノースブリッジ（以下、NB という）63 と、サウスブリッジ（以下、SB という）64 と、ASIC 66 と、ローカルメモリ 67 と、HDD 68、ネットワーク I/F コントローラ 69 とを含む。

【0055】

オペレーションパネル 70 は、コントローラ 60 の ASIC 66 に接続されている。また、FCU 80、USB デバイス 90、IEEE 1394 デバイス 100 およびエンジン部 110 は、コントローラ 60 の ASIC 66 に PCI バスで接続されている。

【0056】

コントローラ 60 は、ASIC 66 にローカルメモリ 67、HDD 68、ネットワーク I/F コントローラ 69 などが接続されると共に、CPU 61 と ASIC 66 とが CPU チップセットの NB 63 を介して接続されている。NB 63 を介して CPU 61 と ASIC 66 とを接続すれば、CPU 61 の I/F が公開されていない場合に対応できる。

【0057】

なお、ASIC 66 と NB 63 とは、PCI バスを介して接続されているのではなく、AGP (Accelerated Graphics Port) 65 を介して接続されている。このように、図 8 のアプリケーション層 5 やプラットフォーム 6 を形成する一つ以上のプロセスを実行制御するため、ASIC 66 と NB 63 とを低速の PCI バスでなく AGP 65 を介して接続し、パフォーマンスの低下を防いでいる。

【0058】

CPU 61 は、融合機 1 の全体制御を行うものである。CPU 61 は、NCS 31、DCS 32、OCS 33、FCS 34、ECS 35、MCS 36、UCS 37、SCS 38、SRM 39、FCUH 40 および IMH 41 を OS 上にそれぞれプロセスとして起動して実行させると共に、アプリケーション層 111 を形成するプリンタアプリ 21、コピーアプリ 22、ファックスアプリ 23、スキャナアプリ 24、ネットファイルアプリ 25、WEB サービス I/F 26 を起動して実行させる。

【0059】

NB 63 は、CPU 61、システムメモリ 62、SB 64 および ASIC 66 を接続するためのブリッジである。システムメモリ 62 は、融合機 1 の描画用メモリ等として用いるメモリである。SB 64 は、図示していない ROM、PCI バス、周辺デバイス等と NB 63 とを接続するためのブリッジである。

【0060】

ローカルメモリ 67 はコピー用画像バッファ、符号バッファとして用いるメモリである。ASIC 66 は、画像処理用のハードウェア要素を有する画像処理用途向けの IC である。HDD 68 は、画像データの蓄積、文書データの蓄積、プログラムの蓄積、フォントデータの蓄積、フォームの蓄積などを行うためのストレージである。また、オペレーションパネル 70 は、ユーザからの入力操作を受け付けると共に、ユーザに向けた表示を行う操作部である。

【0061】

以下、ネットワークを介して接続されたクライアントから文字列を含む要求（文書名や作者名の変更、照合）を受信したときの融合機 1 の処理について説明していく。

【実施例 1】

【0062】

図 10 は、本発明による第 1 実施例の文字列処理方法について説明するための融合機の構成図である。なお、図 10 の構成図は説明に必要な構成を省略している。

【0063】

図 10 では、融合機 1 にインターネット、LAN などのネットワーク 210 を介してクライアント 200 が接続されている。クライアント 200 と融合機 1 とは、XML (eXtensible

nsible Markup Language) 形式でデータ交換を行う。なお、クライアント 200 と融合機 1 とは、例えばデータ・アクセス・プロトコルとして SOAP (Simple Object Access Protocol) と呼ばれる通信プロトコル規格を利用する。

【0064】

ステップ S1 では、クライアント 200 が、XML 形式で記述した図 11 のような SOAP メッセージを作成し、その SOAP メッセージを HTTP メッセージに格納させて融合機 1 に送信する。図 11 は、文書名の変更要求を行う SOAP メッセージの一例の構成図である。クライアント 200 から送信された HTTP メッセージは、NCSS 31 の httpd 220 を介して Web サービス I/F 26 に供給される。

【0065】

HTTP メッセージを受信すると、Web サービス I/F 26 はデイスパッチャにより SOAP メッセージを解釈したあと、文書名の変更要求に応じた set Title のオブジェクト 230 に処理を依頼する。

【0066】

まず、set Title のオブジェクト 230 は、内部処理で利用する符号化方式以外の文字列 (変更後の文書名) を受信すると、その文字列を内部処理で利用する符号化方式に変換する必要がある。ここでは、UTF-8 で符号化された文字列を Shift_JIS または Latin1 で符号化された文字列に変換する例を説明する。

【0067】

Web サービス I/F 26 から処理を依頼されると、オブジェクト 230 は文字列の符号化方式を変換する必要があるかを判定し、必要があれば図 12 のようなフローチャートで表す処理を実行する。

【0068】

図 12 は、符号化方式決定処理の一例のフローチャートである。図 13 は UTF-8, Shift_JIS および Latin1 の文字コードの割り当てを表した一例のルックアップテーブルである。

【0069】

ステップ S11 では、オブジェクト 230 は融合機 1 で利用可能な符号化方式のリストを例えば HDD 68 などから取得し、最初の符号化方式として例えば Shift_JIS を選択する。なお、符号化方式のリストはランダムに並んだものであってもよい、優先度順に並んだものであってもよい。

【0070】

ステップ S11 に続いてステップ S12 に進み、オブジェクト 230 は符号化方式を変更する文字列

【0071】

[外 4]

「 Göte 作 」

をステップ S11 で選択した Shift_JIS で符号化する。図 13 のルックアップテーブルを参照すると、符号化方式 Shift_JIS では文字

【0072】

[外 5]

「
ö」

が変換できない。ステップ S 1 2 に続いてステップ S 1 3 に進み、オブジェクト 2 3 0 はステップ S 1 2 で符号化した結果、符号化方式を変更する文字列がステップ S 1 1 で選択した Shift_JIS で所定基準以上の符号化が可能であるか否かを判定する。例えば所定基準としては、完全な符号化であってもよいし、選択した符号化方式で変換できない文字の数が所定数以下であってもよい。ここでは、所定基準が完全な符号化である例を説明する。

【0073】

この場合、Shift_JIS で完全に符号化が可能ではないので（S 1 3 において NO）、オブジェクト 2 3 0 はステップ S 1 4 に進む。なお、ステップ S 1 1 で選択した符号化方式で完全に符号化が可能であれば（S 1 3 において YES）、オブジェクト 2 3 0 はステップ S 1 1 で選択した符号化方式に決定して処理を終了する。

【0074】

ステップ S 1 4 では、オブジェクト 2 3 0 が、ステップ S 1 2 で符号化した結果から、代用文字に置き換える文字の数および文字の位置を例えばシステムメモリ 6 2 に記憶する。この場合、オブジェクト 2 3 0 は代用文字に置き換える文字の数「1」および文字の位置「最初から 2 番目」を記憶する。

【0075】

ステップ S 1 4 に続いてステップ S 1 5 に進み、オブジェクト 2 3 0 は符号化方式のリストから選択されていない符号化方式があるか否かを判定する。この場合、Latin1 が選択されていないので（S 1 5 において YES）、オブジェクト 2 3 0 はステップ S 1 6 に進む。

【0076】

ステップ S 1 5 に続いてステップ S 1 6 に進み、オブジェクト 2 3 0 は符号化方式のリストから Shift_JIS の次の Latin1 を選択してステップ S 1 2 に戻る。ステップ S 1 2 では、オブジェクト 2 3 0 が符号化方式を変更する文字列

【0077】

[外 6]

「
Göte 作」

をステップ S 1 6 で選択した Latin1 で符号化する。図 1 3 のルックアップテーブルから、符号化方式 Latin1 では文字「作」が変換できないことが分かる。ステップ S 1 2 に続いてステップ S 1 3 に進み、オブジェクト 2 3 0 はステップ S 1 2 で符号化した結果、符号化方式を変更する文字列がステップ S 1 6 で選択した Latin1 で完全に符号化可能であるか否かを判定する。

【0078】

この場合、Latin1 で完全に符号化が可能ではないので（S 1 3 において NO）、オブジェクト 2 3 0 はステップ S 1 4 に進む。ステップ S 1 4 では、オブジェクト 2 3 0 が、ステップ S 1 2 で符号化した結果から、代用文字に置き換える文字の数および文字の

位置を例えばシステムメモリ 6 2 に記憶する。この場合、オブジェクト 2 3 0 は代用文字に置き換える文字の数「1」および文字の位置「最初から 5 番目」を記憶する。

【0 0 7 9】

ステップ S 1 4 に続いてステップ S 1 5 に進み、オブジェクト 2 3 0 は符号化方式のリストから選択されていない符号化方式があるか否かを判定する。この場合、全ての符号化方式が選択されているので（S 1 5 において NO）、オブジェクト 2 3 0 はステップ S 1 8 に進む。

【0 0 8 0】

ステップ S 1 8 では、オブジェクト 2 3 0 が、ステップ S 1 4 で記憶した代用文字に置き換える文字の数、文字の位置などを用いて、所定条件に基づき最適な符号化方式を決定して処理を終了する。

【0 0 8 1】

例えばステップ S 1 8 では、代用文字に置き換える文字の数が一番少ない符号化方式を最適な符号化方式として決定できる。また、代用文字に置き換える文字が連続して出現する回数が一番少ない符号化方式を最適な符号化方式として決定することもできる。また、代用文字に置き換える文字の最初の出現位置が一番後ろの符号化方式を最適な符号化方式として決定することもできる。

【0 0 8 2】

また、代用文字に置き換える文字の平均の出現位置が一番後ろの符号化方式を最適な符号化方式として決定することもできる。また、符号化方式のリスト中で先に現れる符号化方式を最適な符号化方式として決定することもできる。

【0 0 8 3】

さらに、上記のような符号化方式の決定を 2 つ以上組み合わせるか、ポイント制にすることにより、最適な符号化方式を決定することもできる。例えば代用文字に置き換える文字の数が一番少なく、且つ代用文字に置き換える文字の最初の出現位置が一番後ろの符号化方式を最適な符号化方式として決定する場合、

【0 0 8 4】

[外 7]

「 G ö t t e 作 」

を符号化する符号化方式として L a t i n 1 が選択される。なお、代用文字に置き換える文字の数が一番少ない符号化方式を最適な符号化方式として決定すると共に、代用文字に置き換える文字の数が同じ場合に符号化方式のリスト中で先に現れる符号化方式を最適な符号化方式として決定することもできる。

【0 0 8 5】

図 1 2 のフローチャートにより最適な符号化方式を選択したあと、オブジェクト 2 3 0 は選択した符号化方式で文字列

【0 0 8 6】

[外 8]

「 Göte 作 」

を符号化する。オブジェクト 230 は、符号化できない文字を代用文字に置き換える。例えば

【0087】

[外9]

「 Göte 作 」

を Latin1 で符号化した場合、最初から 5 番目の「作」が代用文字に置き換えられる。そして、オブジェクト 230 はステップ S2 に進み、符号化した文字列を用いて MCS 36 に文書名の変更要求を行う。MCS 36 は例えば図 14 のような形式の書誌情報データを HDD 68 等に格納しており、オブジェクト 230 からの文書名の変更要求に応じて書誌情報データを更新する。

【0088】

そして、ステップ S2 に続いてステップ S3 に進み、MCS 36 は文書名の変更応答をオブジェクト 230 に送信する。オブジェクト 230 は文書名の変更応答を受信すると、Web サービス I/F 26 に処理の終了を通知する。

【0089】

ステップ S4 では、Web サービス I/F 26 が、文書名の変更応答を行う SOAP メッセージを XML 形式で記述し、その SOAP メッセージを HTTP メッセージに格納させてクライアント 200 に送信する。

【0090】

以上、本発明による第 1 実施例の文字列処理方法によれば、一の符号化方式で符号化された文字列を他の符号化方式に変換するときに、適切な符号化方式を選択することができる。

【0091】

なお、図 10 では set Title のオブジェクト 230 を Web サービス I/F 26 に設けた例について説明したが、MCS 36 に設けてもよい。MCS 36 にオブジェクト 230 を設けた場合、Web サービス I/F 26 はデイスパッチャにより SOAP メッセージを解釈したあと、文書名の変更要求を MCS 36 に対して行う。そして、MCS 36 は文書名の変更要求に応じた set Title のオブジェクト 230 に処理を依頼する。

【0092】

Web サービス I/F 26 から処理を依頼されると、オブジェクト 230 は文字列の符号化方式を変換する必要があるかを判定し、必要があれば図 12 のようなフローチャートで表す処理により最適な符号化方式を選択する。オブジェクト 230 は選択した符号化方式で文字列を符号化すると共に、符号化できない文字を代用文字に置き換える。そして、オブジェクト 230 はステップ S2 に進み、符号化した文字列を用いて MCS 36 に文書名の変更要求を行う。その他の処理は、set Title のオブジェクト 230 を Web サービス I/F 26 に設けた例と同様である。

【0093】

set Title のオブジェクト 230 を Web サービス I / F 26 に設けた場合のメリットは、第 1 実施例の文字列処理方法が、ネットワーク 210 に接続しない融合機に不要な機能であるため、Web サービス対応のためのオプションという構成にできることである。また、set Title のオブジェクト 230 を MCS 36 に設けた場合のメリットは、他の I / F (例えばプリンタ I / F など) でも利用できることである。

【実施例 2】**【0094】**

第 1 実施例のように、変換できない文字を代用符号に置き換えた場合、変換後の文字列を照合するとき等、代用符号に置き換えた全ての文字が同じ文字としてみなされるという問題もあった。また、符号化方式が等しくないと文字の一致とみなさない場合、同じ文字が異なる文字とみなされるという問題があった。

【0095】

そこで、第 2 実施例の融合機 1 は照合を行う文字列の符号化方式を等しい符号化方式に変換させ、置き換えられた代用符号をワイルドカード扱いとすることにより、上記の問題を解決している。

【0096】

図 15 は、文字列の照合について説明するための図である。例えば UTF-8 で符号化された

【0097】

[外 10]

「 Göte 作 」

を Shift_JIS または Latin1 で符号化し、再び UTF-8 に符号化した文字列は、

【0098】

[外 11]

「 Göte 作 」

に一致していない。そこで、代用符号で表される不明文字をワイルドカード扱いとすることで、

【0099】

[外 12]

「 Göte 作 」

に完全一致する文字列を照合できる。また、

【0100】
[外13]

「Göte」

に前方一致する文字列も照合できる。また、Windows（登録商標）やMAC（登録商標）などのOSの違いにより、UTF-8では同一の文字に異なる文字コードを例えば図16のように割り当てている。このような例外的な文字を予め図16のようなテーブルに設定しておくことで、本発明の文字列処理方法では異なる文字コードを割り当てられた文字であっても同一の文字としてみなすことができる。なお、図16のテーブルはデータの保持形式の一例であって、例えばXMLなどであってもよい。

【0101】

図16は、例外的な文字を設定したテーブルの一例の構成図である。図16のテーブルを利用することで、本発明の文字列処理方法は例外的な文字を含む文字列であっても正確な照合が可能である。なお、本発明の文字列処理方法では大文字および小文字、平仮名および片仮名の関係を例外的な文字と同様、テーブルに設定しておくことで、異なる文字であっても同一の文字としてみなすこともできる。

【0102】

図17は、本発明による第2実施例の文字列処理方法について説明するための融合機の構成図である。なお、図17の構成図は説明に必要な構成を省略している。また、図17の融合機1は図10の融合機1と一部を除いて同様な構成であるため適宜説明を省略する。

【0103】

図17の融合機1は、WebサービスI/F26に文書名の照合要求に応じたSearchのオブジェクト240を備え、また、WebサービスI/F26が利用するキャッシュ250を備えるように構成される。

【0104】

図18は、文字列の照合処理について説明するための一例のフローチャートである。クライアント200から上記の文字列の符号化方式の変換を経て文書名の変更または文書の作成に伴う文書名の設定を指示するか、操作者がオペレーションパネル70を操作して文書名の変更または文書の作成を指示すると、融合機1のQCS33はステップS21に進み、文書名の変更要求または文書の作成要求をMCS36に送信する。

【0105】

ステップS21に続いてステップS22に進み、MCS36は文書名の変更要求または文書の作成要求を受信し、その文書名の変更要求または文書の作成要求に応じて図14のような書誌情報データを変更する。そして、MCS36はファイル変化通知をWebサービスI/F26に送信する。

【0106】

ファイル変化通知を受信すると、WebサービスI/F26は文書名の照合要求に応じたsearchのオブジェクト240に処理を依頼する。図19は、符号化処理の一例のフローチャートである。

【0107】

ステップS31では、受信したファイル変化通知に含まれている文書名が内部処理で利用するShift_JISまたはLatin1等の符号化方式で符号化されているため、UTF-8に変換される。ステップS31に続いてステップS32に進み、オブジェクト240は受信したファイル変化通知に含まれている文書IDをキー情報としてキャッシュ

2 5 0 から該当する文書名を検索して照合する。

【0 1 0 8】

ステップ S 3 2 に続いてステップ S 3 3 に進み、オブジェクト 2 4 0 はステップ S 3 2 での照合結果から、U T F - 8 に変換された文書名とキャッシュ 2 5 0 から検索された文書名とが一致とみなせるか否かを判定する。

【0 1 0 9】

U T F - 8 に変換された文書名とキャッシュ 2 5 0 から検索された文書名とが一致とみなせると判定すると (S 3 3 において Y E S)、オブジェクト 2 4 0 は図 1 9 の符号化処理を終了する。

【0 1 1 0】

例えば U T F - 8 に変換された文書名が

【0 1 1 1】

[外 1 4]

「 G t e 作 」

であり、キャッシュ 2 5 0 から検索された文書名が

【0 1 1 2】

[外 1 5]

「 G ö t e 作 」

である場合、前述したように代用符号で表される不明文字をワイルドカード扱いとすることで、U T F - 8 に変換された文書名とキャッシュ 2 5 0 から検索された文書名とを一致とみなせる。クライアント 2 0 0 からの文書名の設定の際には内部処理で利用する符号化方式に変換する前の文字列をキャッシュ 2 5 0 にまず格納する方式をとれば、クライアント 2 0 0 からの文書名の設定の際に、このような場合が発生する。

【0 1 1 3】

一方、U T F - 8 に変換された文書名とキャッシュ 2 5 0 から検索された文書名とが一致とみなせないと判定すると (S 3 3 において N O)、オブジェクト 2 4 0 は U T F - 8 に変換された文書名を新しい文書名としてキャッシュ 2 5 0 に格納して図 1 9 の符号化処理を終了する。

【0 1 1 4】

例えば U T F - 8 に変換された文書名が

【0 1 1 5】

[外 1 6]

「ゲーテ 作」

であり、キャッシュ 2 5 0 から検索された文書名が

【0 1 1 6】

[外 1 7]

「G ö t e 作」

である場合、U T F - 8 に変換された文書名とキャッシュ 2 5 0 から検索された文書名とを一致とみなせない。そこで、オブジェクト 2 4 0 は

U T F - 8 に変換された文書名

【0 1 1 7】

[外 1 8]

「ゲーテ 作」

を新しい文書名としてキャッシュ 2 5 0 に格納する。クライアント 2 0 0 からの文書名の設定の際には内部処理で利用する符号化方式に変換する前の文字列をキャッシュ 2 5 0 に格納しない方式をとれば、キャッシュ 2 5 0 から検索された文書名が

【0 1 1 8】

[外 1 9]

「G ö t e 作」

ではなく

【0 1 1 9】

[外 2 0]

「 G□t e 作 」

であったりするが、不明文字をワイルドカード扱いとしても

【0120】

[外21]

「 ゲーテ 作 」

とは一致しないので、同様の結果となる。

【0121】

このように、図17の融合機1では文書名の変更があったときに、文書名を内部処理で利用するShift_JISまたはLatin1等からUTF-8に変換している。

【0122】

ステップS23では、クライアント200が、XML形式で記述した図20のようなSOAPメッセージを作成し、そのSOAPメッセージをHTTPメッセージに格納させて融合機1に送信する。図20は、文書名の検索要求を行うSOAPメッセージの一例の構成図である。クライアント200から送信されたHTTPメッセージは、NCSS31のhttpd220を介してWebサービスI/F26に供給される。

【0123】

HTTPメッセージを受信すると、WebサービスI/F26はディスパッチャによりSOAPメッセージを解釈したあと、searchのオブジェクト240に処理を依頼する。

【0124】

searchのオブジェクト240は、UTF-8で符号化された文字列（検索を行う文書名）を受信すると、その文字列をキー情報としてキャッシュ250から該当する文書名を検索して照合する。オブジェクト240は文字列の照合処理が終了すると、WebサービスI/F26に処理の終了を通知する。

【0125】

ステップS24では、WebサービスI/F26が、文書名の検索応答を行うSOAPメッセージをXML形式で記述し、そのSOAPメッセージをHTTPメッセージに格納させてクライアント200に送信する。

【0126】

以上、本発明による第2実施例の文字列処理方法によれば、変換後の文字列を照合するときに、符号化方式の違いによる変換後の文字列の差異を吸収することができる。また、キャッシュ250を利用することにより、変換後の文字列を照合するときに、いちいちMCS36に問い合わせる必要がない。更に、クライアント200からの文書名の設定の際には内部処理で利用する符号化方式に変換する前の文字列をキャッシュ250にまず格納する方式を取っておけば、クライアント200からの文書名の検索の際には不明文字を含まない変換前の文字列同士の照合となるので、不明文字をワイルドカード扱いとするよりも検索の精度が高いものとなる。

【0127】

なお、図17ではsearchのオブジェクト240をWebサービスI/F26に設けた例について説明したが、MCS36に設けてもよい。図21は、文字列の照合処理について説明するための一例のフローチャートである。

【0128】

クライアント200から上記の文字列の符号化方式の変換を経て文書名の変更または文書の作成に伴う文書名の設定を指示するか、操作者がオペレーションパネル70を操作して文書名の変更または文書の作成を指示すると、融合機1のOCS33はステップS21に進み、文書名の変更要求または文書の作成要求をMCS36に送信する。

【0129】

ステップS21に続いてステップS22に進み、MCS36は文書名の変更要求または文書の作成要求を受信し、その文書名の変更要求または文書の作成要求に応じて図14のような書誌情報データを変更する。そして、MCS36は文書名の照合要求に応じたsearchのオブジェクト240に処理を依頼する。

【0130】

オブジェクト240は図19のような符号化処理を行い、UTF-8に変換された文書名とキャッシュ250から検索された文書名とが一致とみなせないと判定したときに、UTF-8に変換された文書名を新しい文書名としてキャッシュ250に格納する。

【0131】

このように、図17の融合機1では文書名の変更があったときに、文書名を内部処理で利用するShift_JISまたはLatin1等からUTF-8に変換している。

【0132】

ステップS42では、クライアント200が、XML形式で記述した図20のようなSOAPメッセージを作成し、そのSOAPメッセージをHTTPメッセージに格納させて融合機1に送信する。クライアント200から送信されたHTTPメッセージは、NCS31のhttpd220を介してWebサービスI/F26に供給される。

【0133】

HTTPメッセージを受信すると、WebサービスI/F26はディスパッチャによりSOAPメッセージを解釈したあと、ステップS43で文書名の検索要求をMCS36に送信する。MCS36は文書名の検索要求を受信すると、searchのオブジェクト240に処理を依頼する。

【0134】

searchのオブジェクト240は、UTF-8で符号化された文字列（検索を行う文書名）を受信すると、その文字列をキー情報としてキャッシュ250から該当する文書名を検索して照合する。オブジェクト240は文字列の照合処理が終了すると、MCS36に処理の終了を通知する。

【0135】

ステップS44では、MCS36が、文書名の検索結果をWebサービスI/F26に送信する。ステップS45では、WebサービスI/F26が、文書名の検索応答を行うSOAPメッセージをXML形式で記述し、そのSOAPメッセージをHTTPメッセージに格納させてクライアント200に送信する。

【0136】

以上、本発明による第2実施例の文字列処理方法によれば、変換後の文字列を照合するときに、符号化方式の違いによる変換後の文字列の差異を吸収することができる。

【0137】

Searchのオブジェクト240をWebサービスI/F26に設けた場合のメリットは、第2実施例の文字列処理方法が、ネットワーク210に接続しない融合機に不要な機能であるため、Webサービス対応のためのオプションという構成にできることである。また、Searchのオブジェクト240をMCS36に設けた場合のメリットは、他のI/F（例えばプリンタI/Fなど）でも利用できることである。

【0138】

前述した第1実施例の融合機1ではキャッシュ250を有していないが、図17の融合機1と同様に、WebサービスI/F26またはMCS36が利用するキャッシュ250を備えるようにしてもよい。

【0139】

第1実施例の融合機1でキャッシュ250を備えることにより、変換前の文字列をキャッシュ250に格納しておくことができる。このように、変換前の文字列をキャッシュ250に格納しておくことにより、変換前の文字列を文字列の照合処理に利用することができる。文字列の照合処理の精度を上げることができる。

【実施例3】

【0140】

上記の実施例1では、符号化方式の優先順位が固定されている例を説明したが、様々な条件に応じて符号化方式の優先順位が変化するようにしてもよい。そこで、実施例3では様々な条件に応じて符号化方式の優先順位が変化する例を説明する。

【0141】

図22は、本発明による第3実施例の文字列処理方法について説明するための融合機の構成図である。図23は、本発明による第3実施例の文字列処理方法について説明するためのシーケンス図である。なお、図22及び図23では説明に必要な構成を省略している。また、図22の構成図は図10の構成図と一部を除いて同様であるため、適宜説明を省略する。

【0142】

ステップS51では、クライアント200のブラウザ201が、URLを指定してhttpリクエストを行う。例えばステップS51で指定されるURLは、「http://111.222.33.44/ja/documentbox/」等である。このURLには、ブラウザ201の表示言語が日本語である例であり、URL中の「ja」で表示言語が日本語に指定されている。ブラウザ201から送信されたHTTPリクエストは、融合機1のNCS31に含まれるhttpd220に供給される。

【0143】

ステップS52に進み、httpd220は供給されたhttpリクエストをWebアプリ27に供給する。httpリクエストを受信すると、Webアプリ27はhttpリクエストで指定されているURLからブラウザ201の表示言語を特定し、その表示言語に応じた画面データを生成する。

【0144】

ステップS53に進み、Webアプリ27は生成した画面データをhttpd220に供給する。ステップS54に進み、httpd220は供給された画面データをクライアント200のブラウザ201に送信する。

【0145】

画面データを受信すると、ブラウザ201はステップS55に進み、画面データに応じた画面を表示する。そして、ユーザはブラウザ201に表示された画面に文字列を入力する。ステップS56に進み、クライアント200のブラウザ201はステップS55で画面に入力された文字列を含むhttpリクエストを行う。ブラウザ201から送信されたhttpリクエストは、融合機1のhttpd220に供給される。

【0146】

ステップS57に進み、httpd220は供給されたhttpリクエストをWebアプリ27に供給する。httpリクエストを受信すると、Webアプリ27はステップS58に進み、httpリクエストに含まれる文字列を取得してsetTitleのオブジェクト230に処理を依頼する。

【0147】

まず、setTitleのオブジェクト230は、内部処理で利用する符号化方式以外の文字列を受信した場合、その文字列を内部処理で利用する符号化方式に変換する必要がある。そこで、オブジェクト230は文字列の符号化方式を変換する必要があるかを判定

し、必要があれば前述した図12のようなフローチャートで表す処理を実行する。

【0148】

なお、実施例3の場合、図12のステップS11の処理が実施例1の場合と異なる。オブジェクト230は、符号化方式の優先順位を次のように決定し、その優先順位に従って符号化方式を選択する。優先順位の決定は、次のように行う。まず、ブラウザ201の表示言語に対応する符号化方式は、上位の優先順位とする。オペレーションパネル70の表示言語に対応する符号化方式は、中位の優先順位とする。そして、それ以外の符号化方式は、下位の優先順位とする。この場合の優先順位は、ブラウザ201の表示言語に対応する符号化方式、オペレーションパネル70の表示言語に対応する符号化方式、それ以外の符号化方式となる。

【0149】

ステップS11以外の処理は、実施例1の場合と同様であるため説明を省略する。図12のフローチャートにより最適な符号化方式を選択したあと、オブジェクト230は選択した符号化方式で文字列を符号化する。なお、オブジェクト230は、符号化できない文字を代用文字に置き換える。

【0150】

そして、ステップS59に進み、Webアプリ27はステップS57のhttpリクエストに対応する画面データを作成し、その画面データをhttpd220に供給する。ステップS54に進み、httpd220は供給された画面データをクライアント200のブラウザ201に送信する。

【0151】

以上、本発明による第3実施例の文字列処理方法によれば、一の符号化方式で符号化された文字列を他の符号化方式に変換するときに、様々な条件に応じて符号化方式の優先順位を変化させることができるので、適切な符号化方式を選択することができる。

【0152】

なお、図22ではsetTitleのオブジェクト230をWebアプリ27に設けた例について説明したが、MCS36に設けてもよい。setTitleのオブジェクト230をWebアプリ27に設けた場合のメリットは、実施例3の文字列処理方法が、ネットワーク210に接続しない融合機に不要な機能であるため、Webアプリ27のオプションという構成にできることである。また、setTitleのオブジェクト230をMCS36に設けた場合のメリットは、他のアプリ（例えばプリンタアプリなど）でも利用できることである。

【0153】

本発明は、具体的に開示された実施例に限定されるものではなく、特許請求の範囲から逸脱することなく、種々の変形や変更が可能である。

【0154】

例えば、請求項1記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の数情報に基づき前記代用符号の数が最も少ない符号化方式を前記他の符号化方式として決定することを特徴としてもよい。

【0155】

また、請求項1記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の数情報に基づき前記代用符号が連続して出現する回数が最も少ない符号化方式を前記他の符号化方式として決定することを特徴としてもよい。

【0156】

また、請求項1記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の位置情報に基づき前記代用符号の出現位置が最も後ろの符号化方式を前記他の符号化方式として決定することを特徴としてもよい。

【0157】

また、請求項1記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の位置情報に基づき前記代用符号の平均出現位置が最も後ろの符号化方式を前記他の符号化方

式として決定することを特徴としてもよい。

【0158】

また、上記に記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の数情報および位置情報に基づき、前記代用符号の数、前記代用符号が連続して出現する回数、前記代用符号の出現位置、前記代用符号の平均出現位置の組み合わせから、前記他の符号化方式を決定することを特徴としてもよい。

【0159】

また、上記に記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の数、前記代用符号が連続して出現する回数、前記代用符号の出現位置、前記代用符号の平均出現位置を数値化して、前記数値により前記他の符号化方式を決定することを特徴としてもよい。

【0160】

また、上記に記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の数情報および位置情報に基づき、前記代用符号の数、前記代用符号が連続して出現する回数、前記代用符号の出現位置、前記代用符号の平均出現位置の何れかが同じであるときに、前記複数の符号化方式のリスト中で先に現れる符号化方式を前記他の符号化方式として決定することを特徴としてもよい。

【0161】

また、上記に記載の文字列処理装置の前記符号化方式決定手段は、前記代用符号の数情報および位置情報に基づき、前記代用符号の数、前記代用符号が連続して出現する回数、前記代用符号の出現位置、前記代用符号の平均出現位置の何れかが同じであるときに、前記複数の符号化方式の優先度に応じて前記他の符号化方式を決定することを特徴としてもよい。

【0162】

また、上記に記載の文字列処理装置の前記符号化方式決定手段は、前記選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したとき、前記代用符号が出現しなければ、前記選択している符号化方式を前記他の符号化方式として決定することを特徴としてもよい。

【図面の簡単な説明】

【0163】

【図1】符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。

【図2】符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。

【図3】符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。

【図4】符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。

【図5】符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。

【図6】符号化方式で表現可能な文字の集合が符号化方式ごとに異なっていることを表す図である。

【図7】文字に割り当てられている文字コードを符号化方式ごとに表したルックアップテーブルである。

【図8】本発明による融合機の一実施例の構成図である。

【図9】本発明による融合機の一実施例のハードウェア構成図である。

【図10】本発明による第1実施例の文字列処理方法について説明するための融合機の構成図である。

【図11】文書名の変更要求を行うSOAPメッセージの一例の構成図である。

【図12】符号化方式決定処理の一例のフローチャートである。

【図13】UTF-8, Shift_JISおよびLatin1の文字コードの割り当てを表した一例のルックアップテーブルである。

【図14】書誌情報データの形式を表した図である。

【図15】文字列の照合について説明するための図である。

【図16】例外的な文字を設定したテーブルの一例の構成図である。

【図17】本発明による第2実施例の文字列処理方法について説明するための融合機の構成図である。

【図18】文字列の照合処理について説明するための一例のフローチャートである。

【図19】符号化処理の一例のフローチャートである。

【図20】文書名の検索要求を行うSOAPメッセージの一例の構成図である。

【図21】文字列の照合処理について説明するための一例のフローチャートである。

【図22】本発明による第3実施例の文字列処理方法について説明するための融合機の構成図である。

【図23】本発明による第3実施例の文字列処理方法について説明するためのシーケンス図である。

【符号の説明】

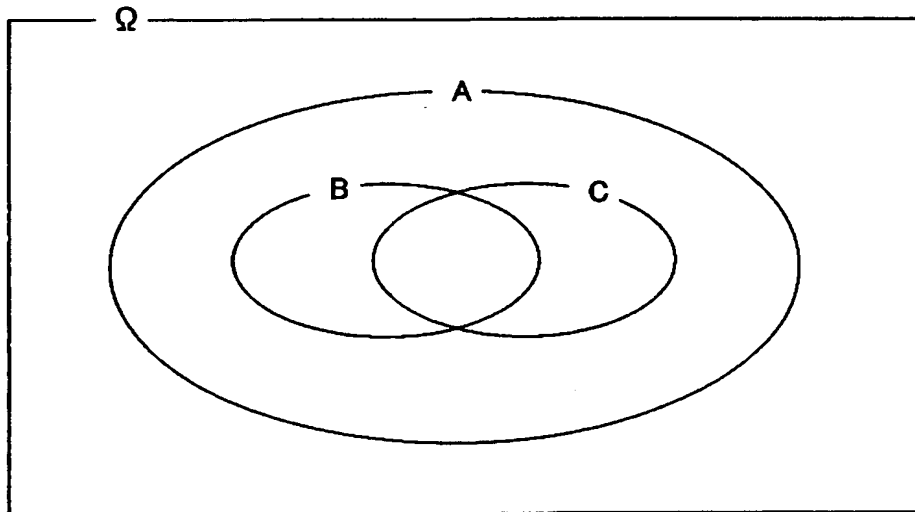
【0164】

- 1 融合機
- 2 ソフトウェア群
- 3 融合機起動部
- 4 ハードウェア資源
- 5 アプリケーション層
- 6 プラットフォーム
- 9 コントロールサービス層
- 10 ハンドラ層
- 11 白黒レーザプリンタ (B&W LP)
- 12 カラーレーザプリンタ (Color LP)
- 13 ハードウェアリソース
- 21 プリンタアプリ
- 22 コピーアプリ
- 23 ファックスアプリ
- 24 スキャナアプリ
- 25 ネットファイルアプリ
- 26 WebサービスI/F
- 27 Webアプリ
- 31 ネットワークコントロールサービス (NCS)
- 32 デリバリーコントロールサービス (DCS)
- 33 オペレーションパネルコントロールサービス (OCS)
- 34 ファックスコントロールサービス (FCS)
- 35 エンジンコントロールサービス (ECS)
- 36 メモリコントロールサービス (MCS)
- 37 ユーザインフォメーションコントロールサービス (UCS)
- 38 システムコントロールサービス (SCS)
- 39 システムリソースマネージャ (SRM)
- 40 ファックスコントロールユニットハンドラ (FCUH)
- 41 イメージメモリハンドラ (IMH)
- 51 アプリケーションプログラムインターフェース (API)
- 52 エンジンI/F
- 60 コントローラ
- 61 CPU

6 2 システムメモリ
6 3 ノースブリッジ (NB)
6 4 サウスブリッジ (SB)
6 5 A G P (Accelerated Graphics Port)
6 6 A S I C
6 7 ローカルメモリ
6 8 ハードディスク装置 (HDD)
6 9 ネットワーク I / F コントローラ
7 0 オペレーションパネル
8 0 ファックスコントロールユニット (FCU)
9 0 U S B デバイス
1 0 0 I E E E 1 3 9 4 デバイス
1 1 0 エンジン部
2 0 0 クライアント
2 0 1 ブラウザ
2 1 0 ネットワーク
2 2 0 h t t p d (HyperText Transfer Protocol Daemon)
2 3 0 s e t T i t l e のオブジェクト
2 4 0 S e a r c h のオブジェクト

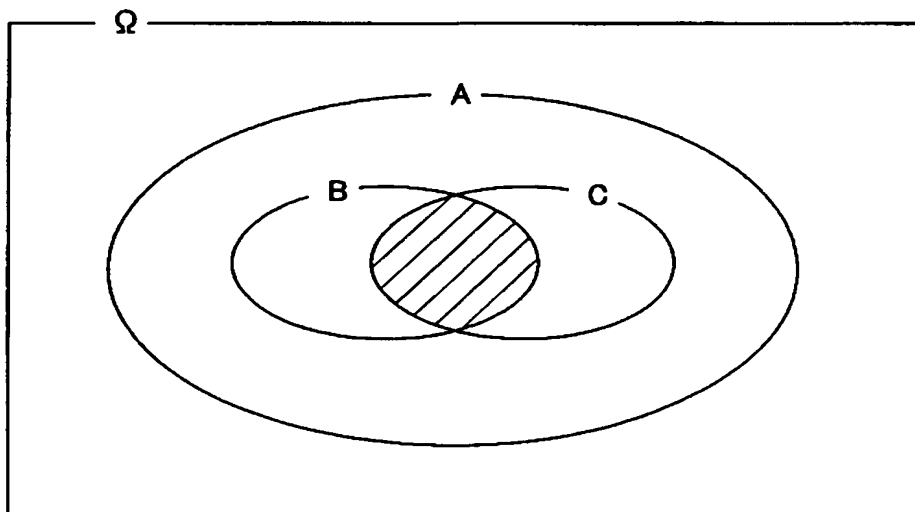
【書類名】 図面
【図 1】

符号化方式で表現可能な文字の集合が
符号化方式ごとに異なっていることを表す図



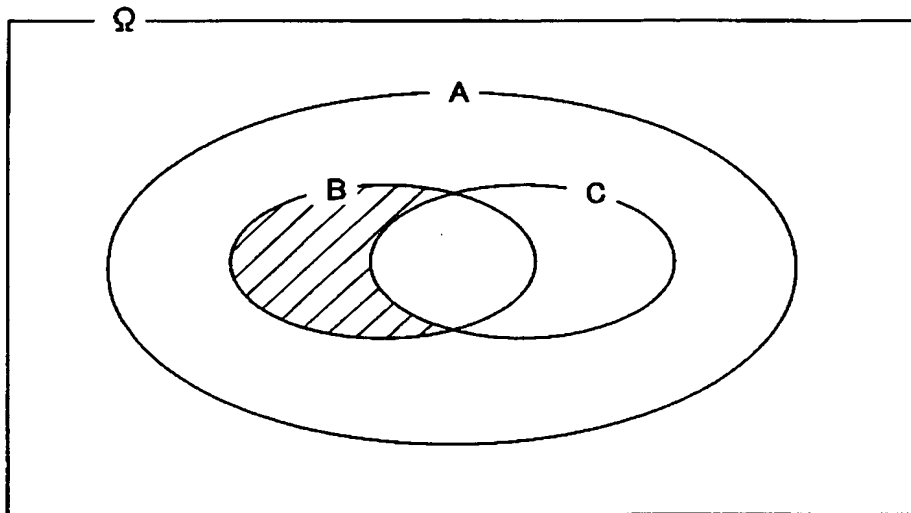
【図 2】

符号化方式で表現可能な文字の集合が
符号化方式ごとに異なっていることを表す図



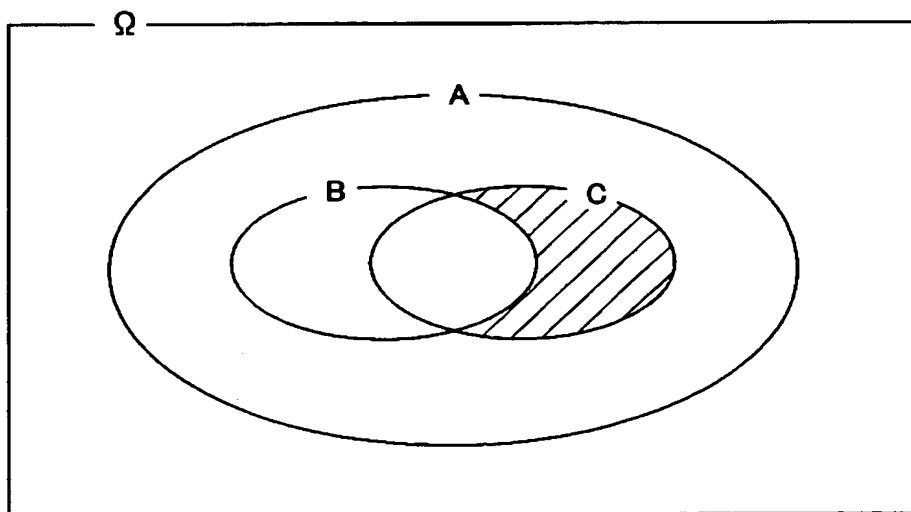
【図 3】

符号化方式で表現可能な文字の集合が
符号化方式ごとに異なっていることを表す図



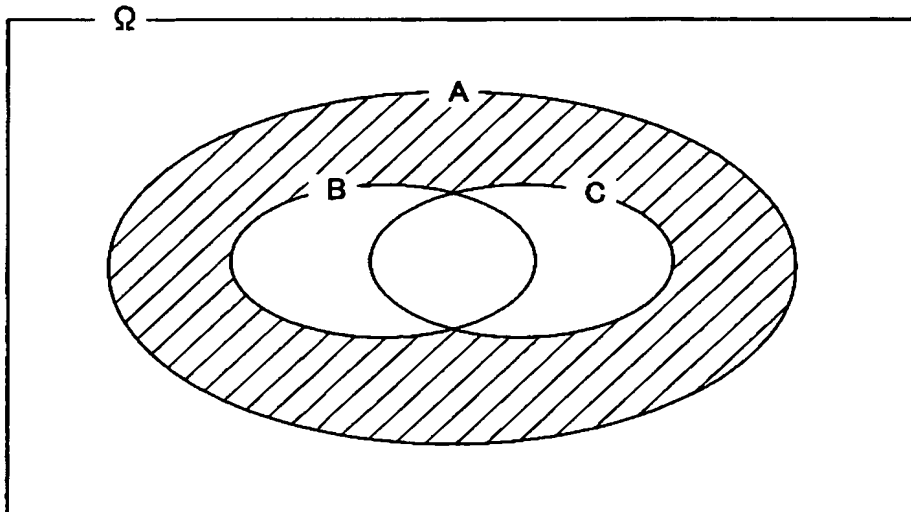
【図 4】

符号化方式で表現可能な文字の集合が
符号化方式ごとに異なっていることを表す図



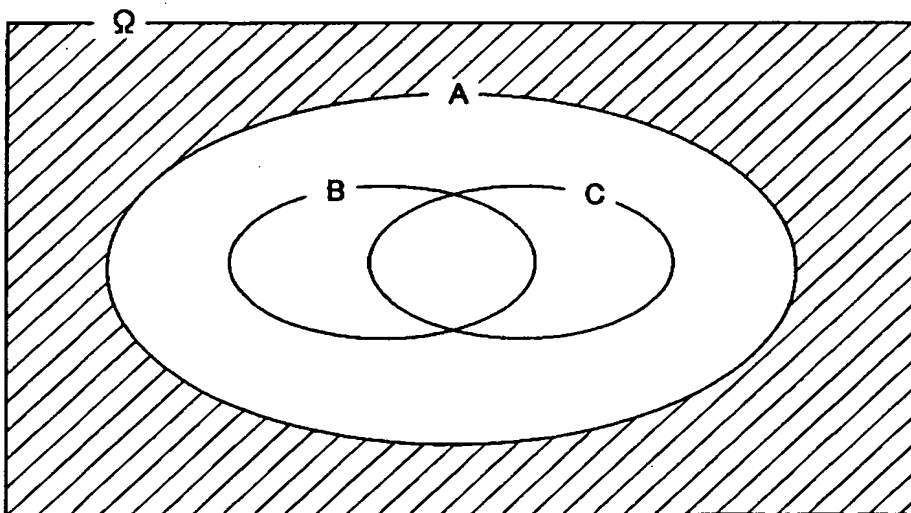
【図 5】

符号化方式で表現可能な文字の集合が
符号化方式ごとに異なっていることを表す図



【図 6】

符号化方式で表現可能な文字の集合が
符号化方式ごとに異なっていることを表す図



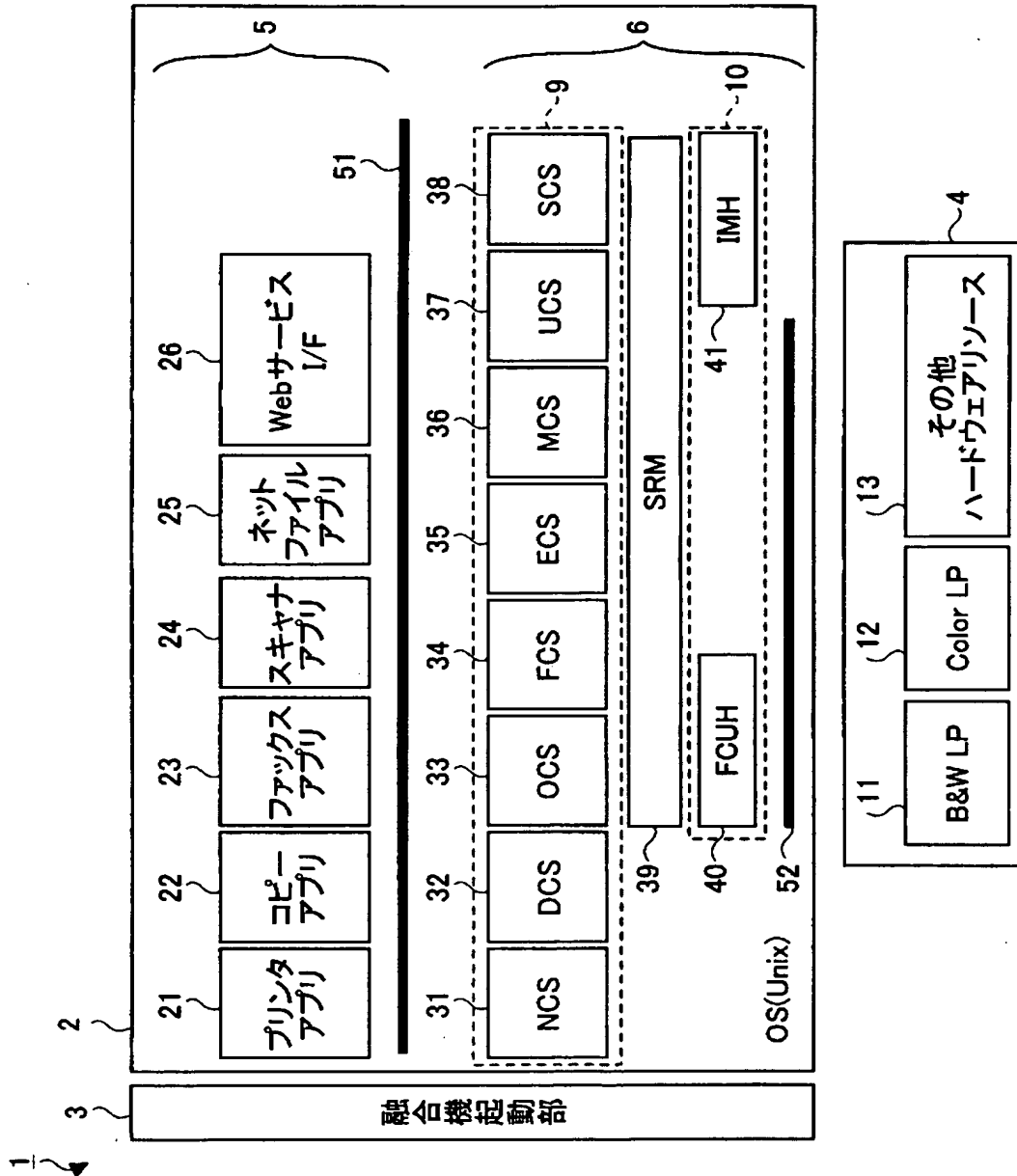
【図 7】

文字に割り当てられている文字コードを
符号化方式ごとに表したルックアップテーブル

文字	UTF-8	Shift_JIS	Latin1
...
1	31	31	31
...
€	E2 82 AC		80
...
Ok	E3 86 AC		
...
円	E5 86 86	89 7E	
...

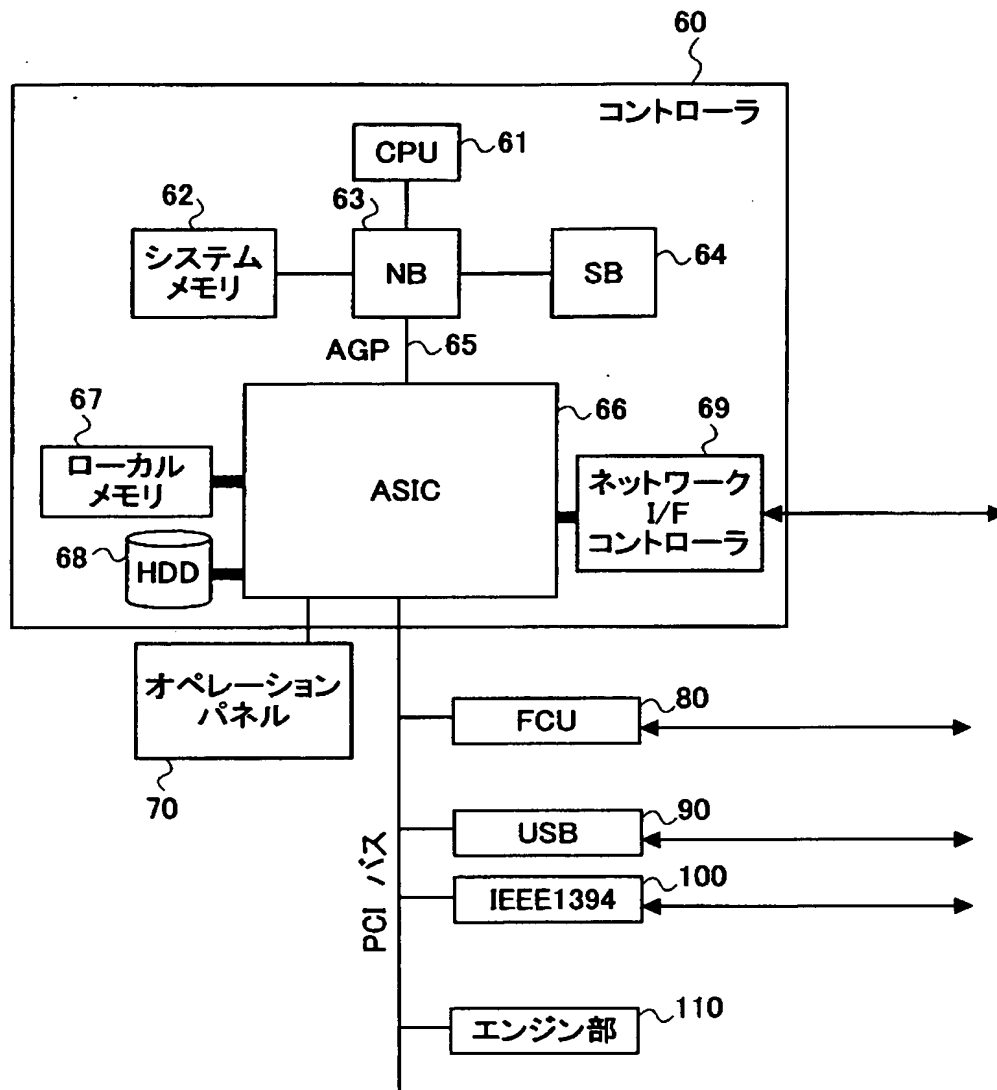
【図 8】

本発明による融合機の一実施例の構成図



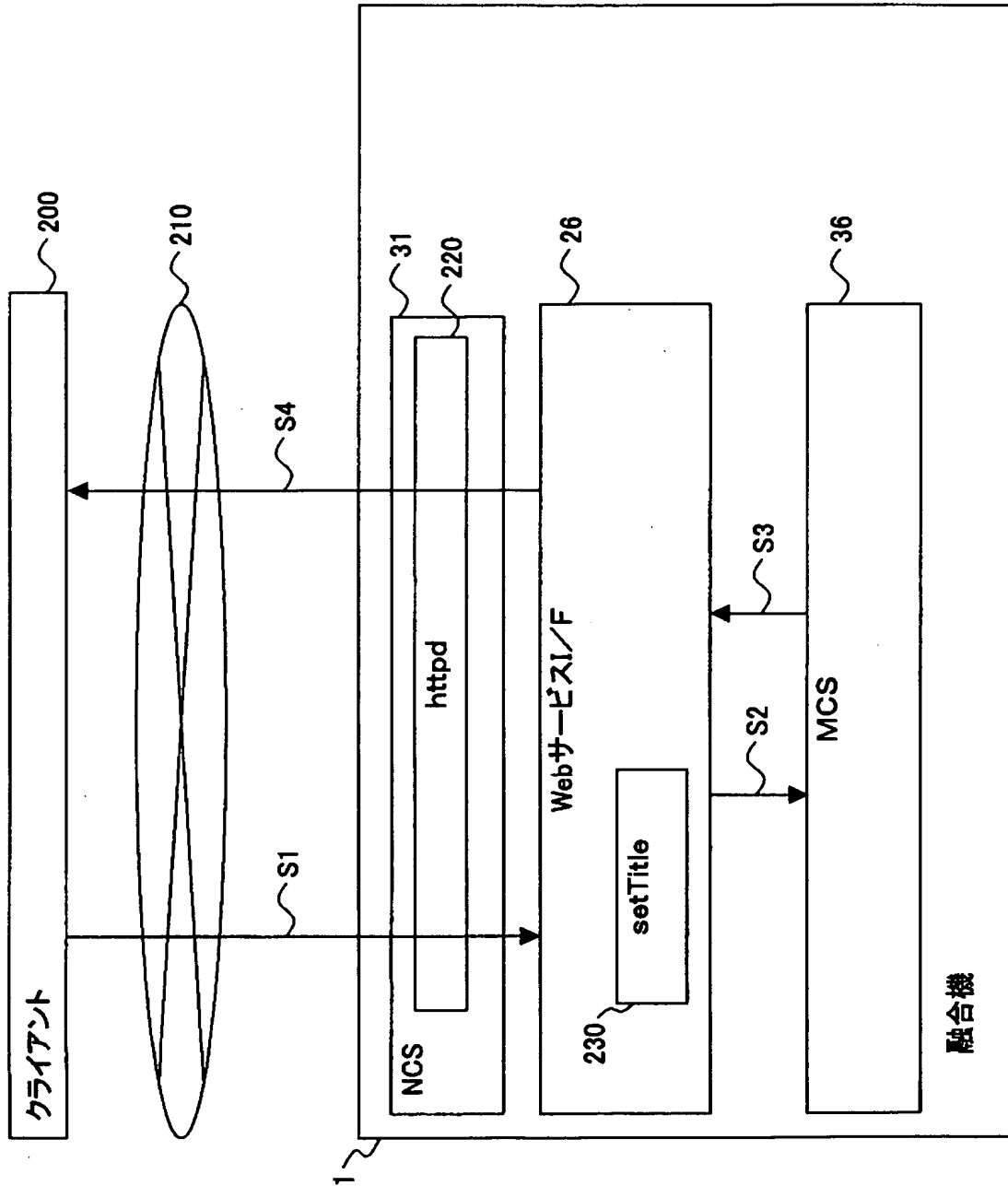
【図 9】

本発明による融合機の一実施例のハードウェア構成図



【図 10】

本発明による第1実施例の文字列処理方法
について説明するための融合機の構成図



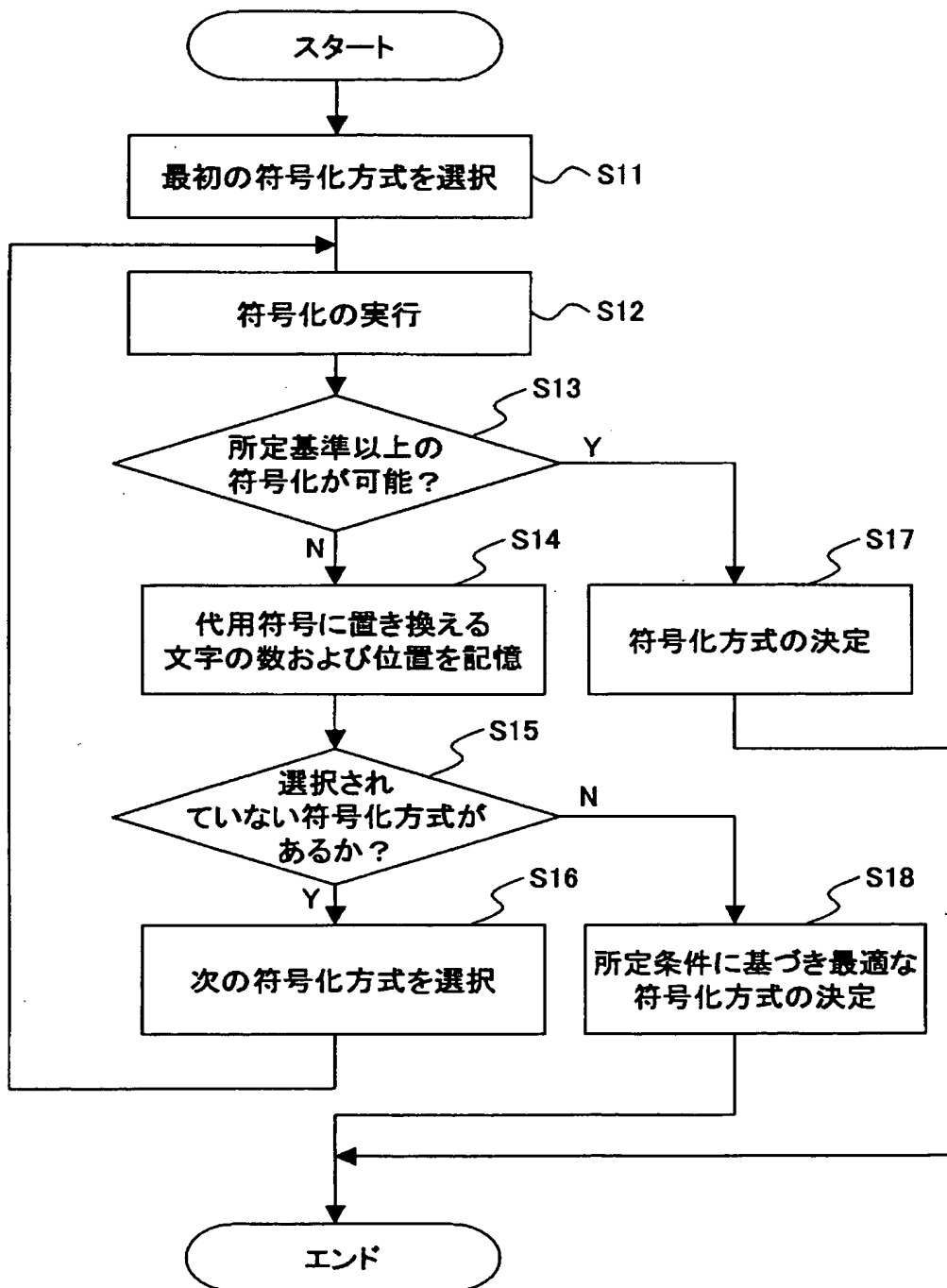
【図 11】

文書名の変更要求を行う SOAP メッセージの一例の構成図

```
- <SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:m0="http://schemas.ricoh.co.jp/commonTypes">
- <SOAP-ENV:Body>
  - <m:setTitle xmlns:m="http://soapdefs.ricoh.co.jp/repository">
    <id xsi:type="xsd:anyURI">doc1234</id>
    <value xsi:type="xsd:string">Göte作</value>
  </m:setTitle>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

【図 12】

符号化方式決定処理の一例のフローチャート



【図 13】

UTF-8, Shift_JISおよびLatin1の
文字コードの割り当てを表した一例のルックアップテーブル

文字	UTF-8	Shift_JIS	Latin1
...
G	47	47	47
...
ö	C3 B6		F6
...
作	E4 BD 9C	8D EC	
...
Ok	E3 86 82		
...

【図 1 4】

書誌情報データの形式を表した図

文書ID	文書名の 符号化方式ID	文書名の 文字列データ	作者名の 符号化方式ID	作者名の 文字列データ	その他の 情報
------	-----------------	----------------	-----------------	----------------	------------

【図 1 5】

文字列の照合について説明するための図

元の文字列 (UTF-8)	変換後の文字列 (内部符号化方式)	再変換後の文字列 (UTF-8)
G ö t e 作	G [?] t e 作 (Shift_JIS)	G [?] t e 作
G ö t e 作	G ö t e [?] (Latin1)	G ö t e [?]

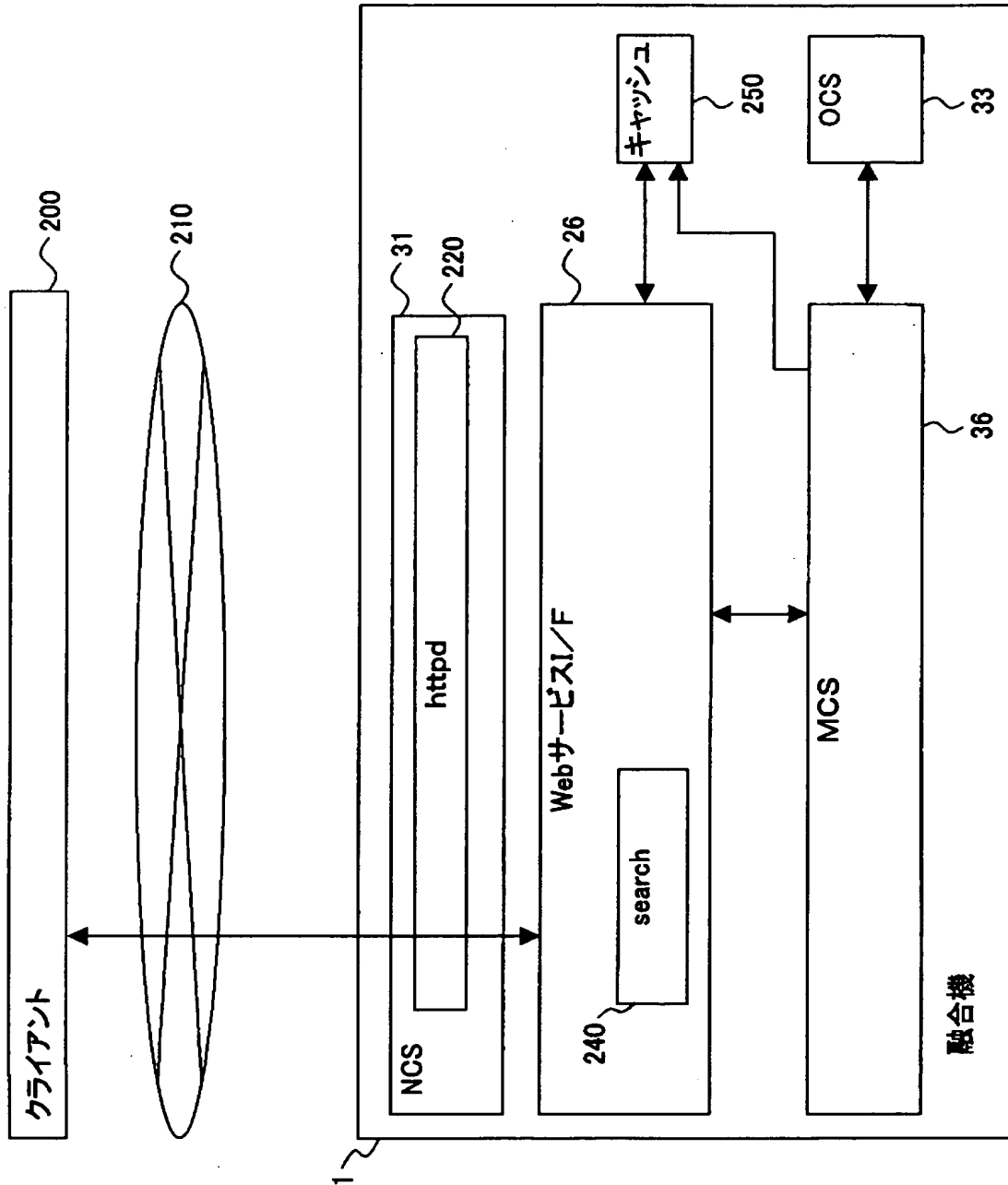
【図 16】

例外的な文字を設定したテーブルの一例の構成図

Shift_JIS		Windows流UTF-8		Mac流UTF-8	
文字	符号	文字名	符号	文字名	符号
¥	5C	バックスラッシュ	5C	円通貨記号	C2 A5
～	7E	テイルダ	7E	オーバーライン	E2 80 BE
～	81 60	全角テイルダ	EF BD 9E	波線	E3 80 9C
＝	81 61	平行線記号	E2 88 A5	双柱記号	E2 80 96
＝	81 7C	全角ハイフン	EF BC 8D	マインス記号	E2 88 92
¢	81 91	全角セント通貨記号	EF BF A0	セント通貨記号	C2 A2
£	81 92	全角ポンド通貨記号	EF BF A1	ポンド通貨記号	C2 A3
ー	81 CA	全角否定記号	EF BF A2	否定記号	C2 AC

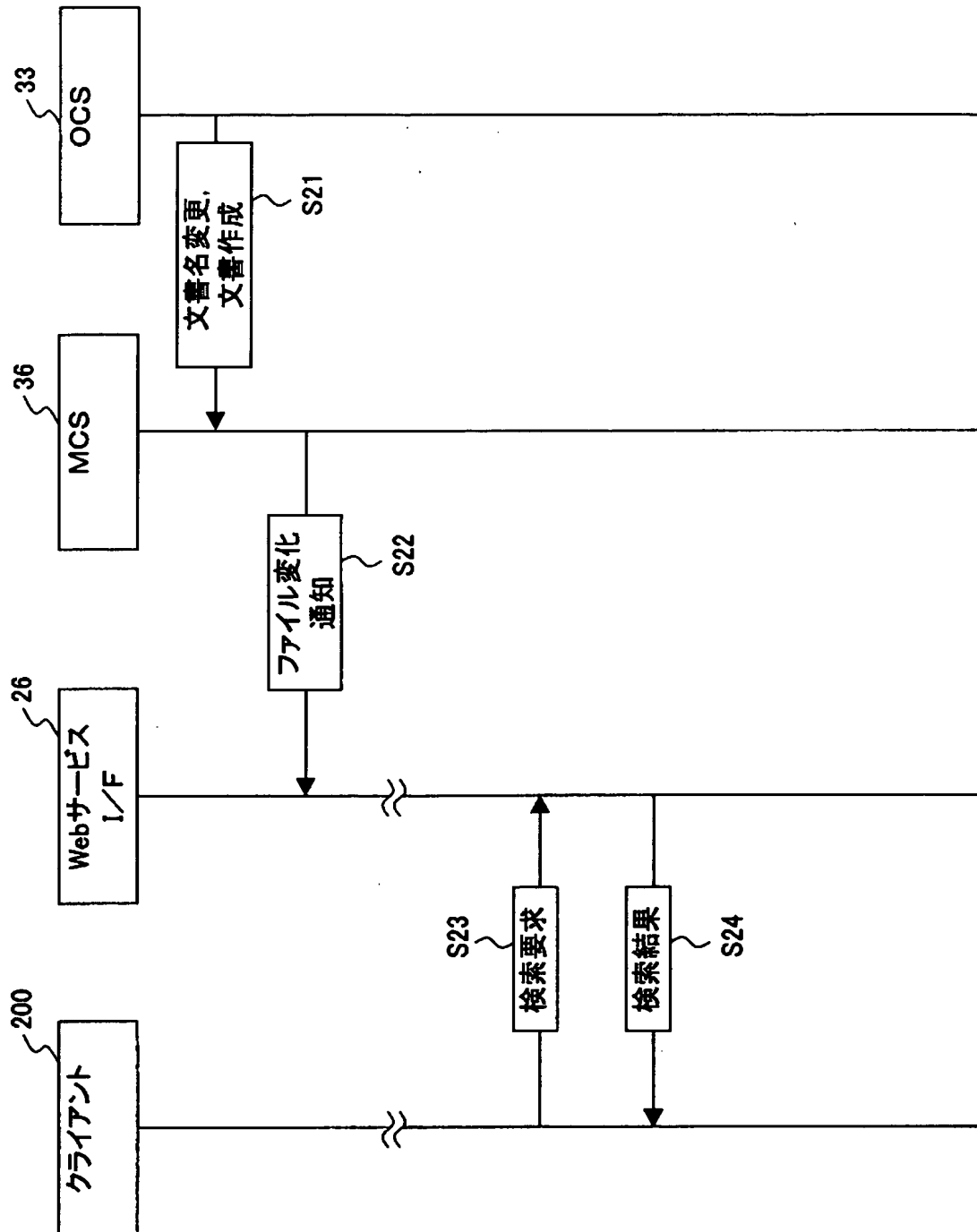
【図 17】

本発明による第2実施例の文字列処理方法
について説明するための融合機の構成図



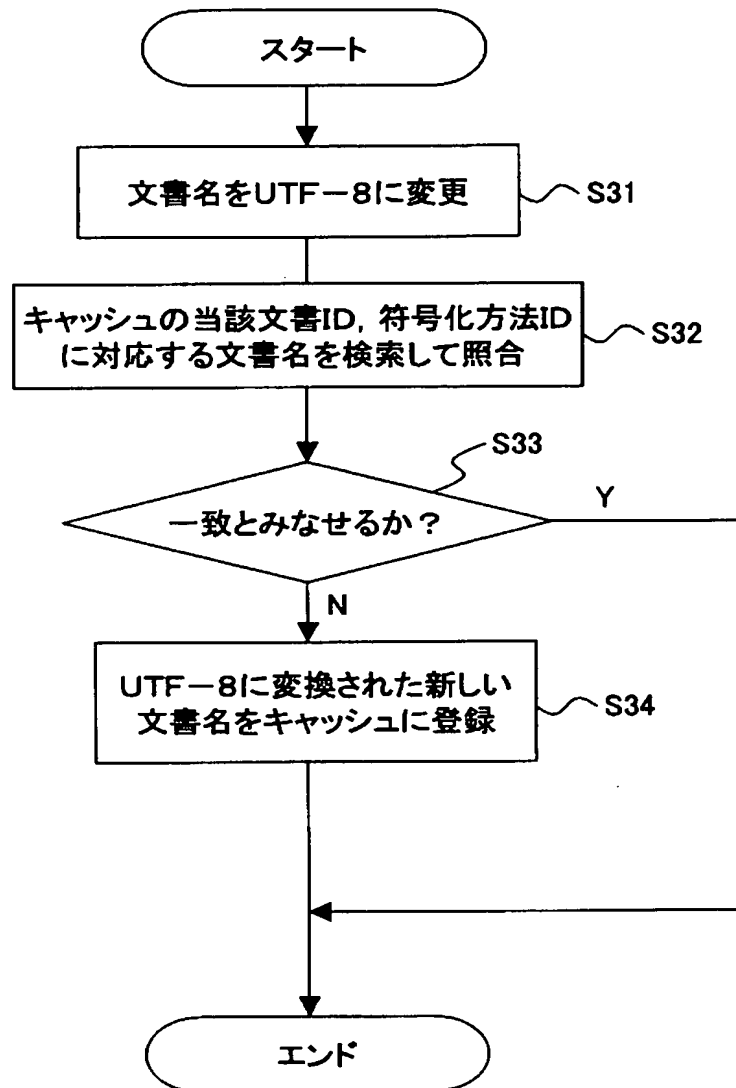
【図18】

文字列の照合処理について説明するための一例のフローチャート



【図 19】

符号化処理の一例のフローチャート



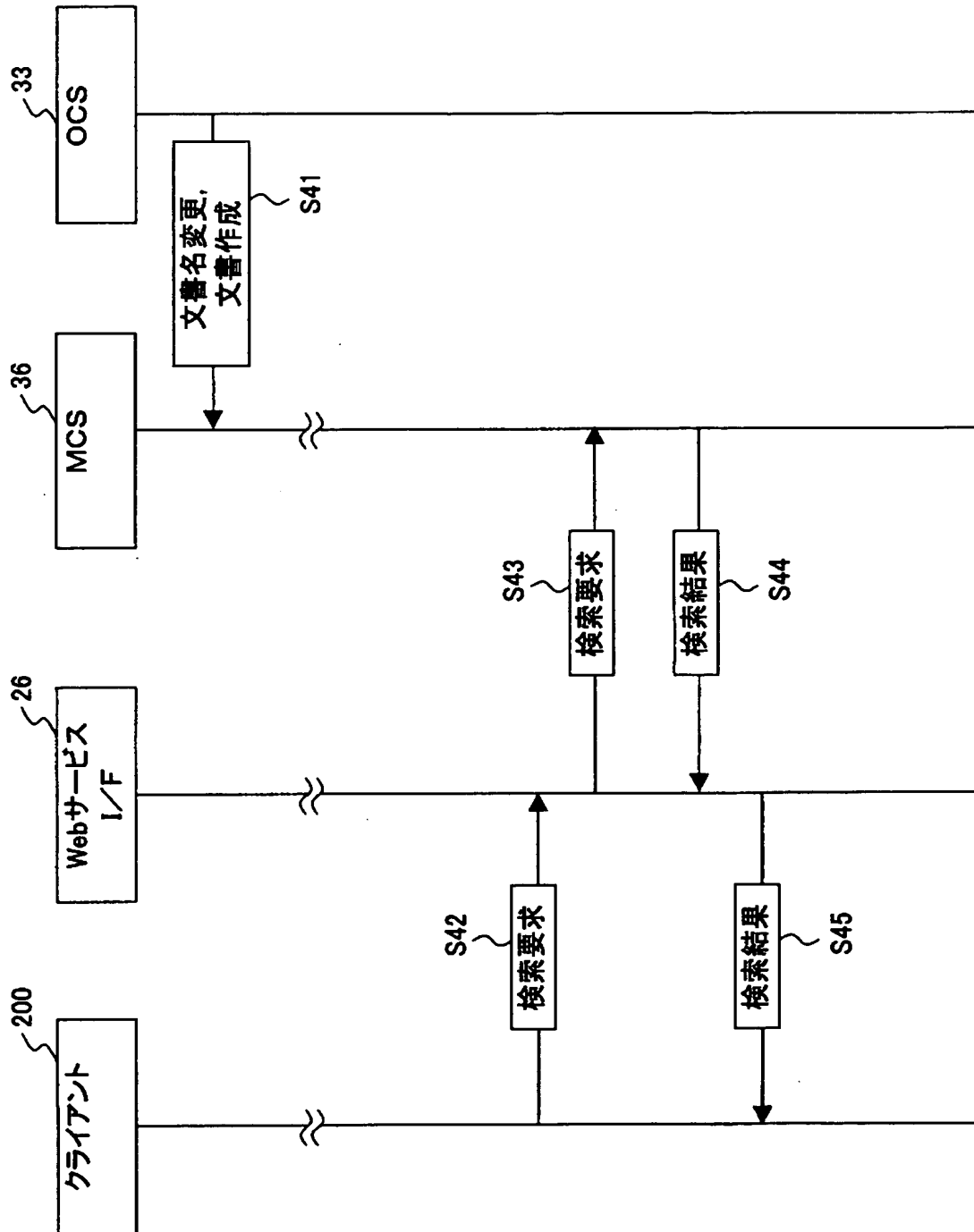
【図 20】

文書名の検索要求を行う SOAP メッセージの一例の構成図

```
- <SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:m0="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:m1="http://schemas.ricoh.co.jp/repository">
- <SOAP-ENV:Body>
- <m:search xmlns:m="http://soapdefs.ricoh.co.jp/repository">
  - <condition>
    <operator>CONTAINS_ANY_AT_HEAD</operator>
    <propName>title</propName>
    <propVal>Göte</propVal>
  </condition>
  </m:search>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

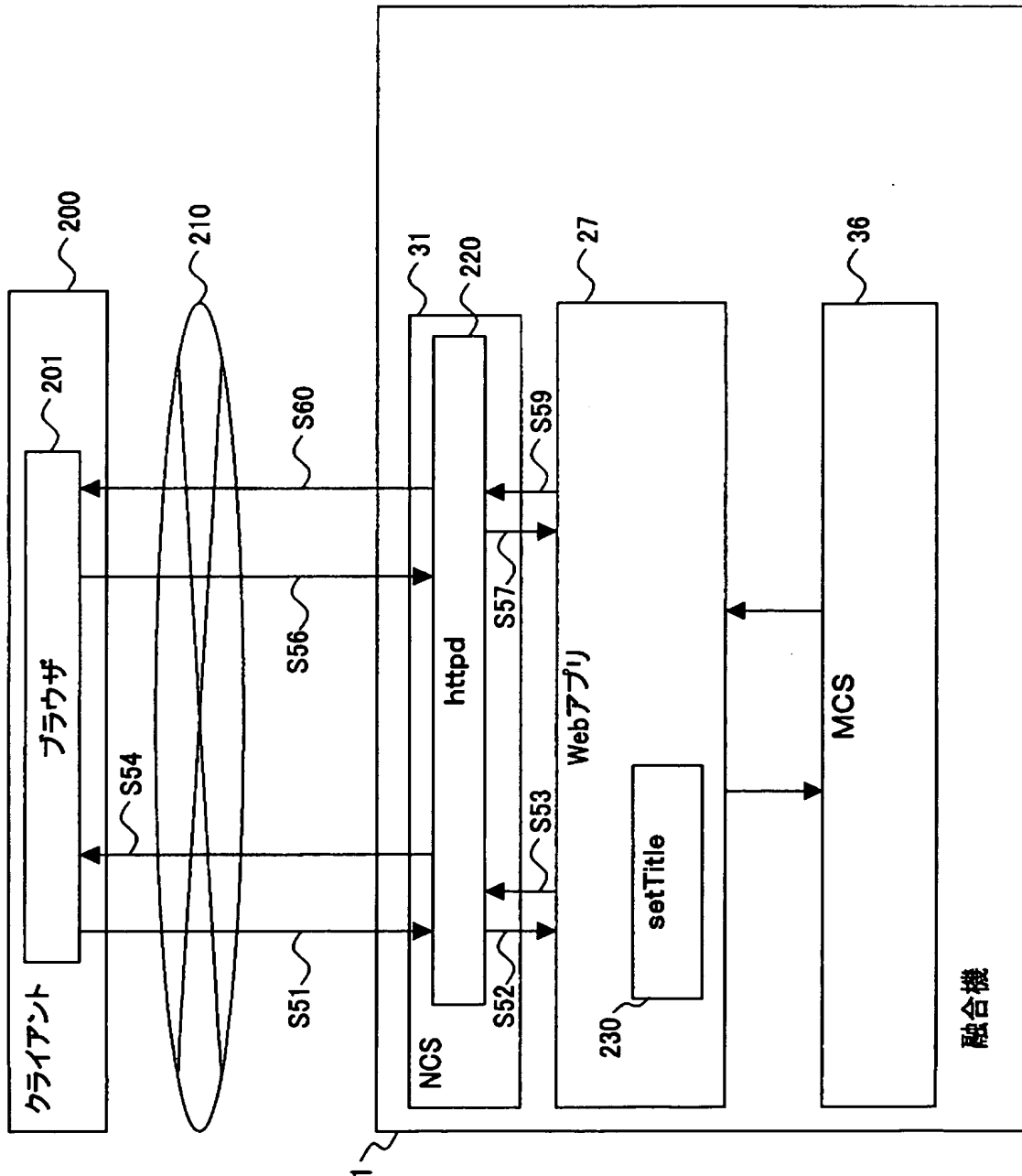
【図 21】

文字列の照合処理について説明するための一例のフローチャート

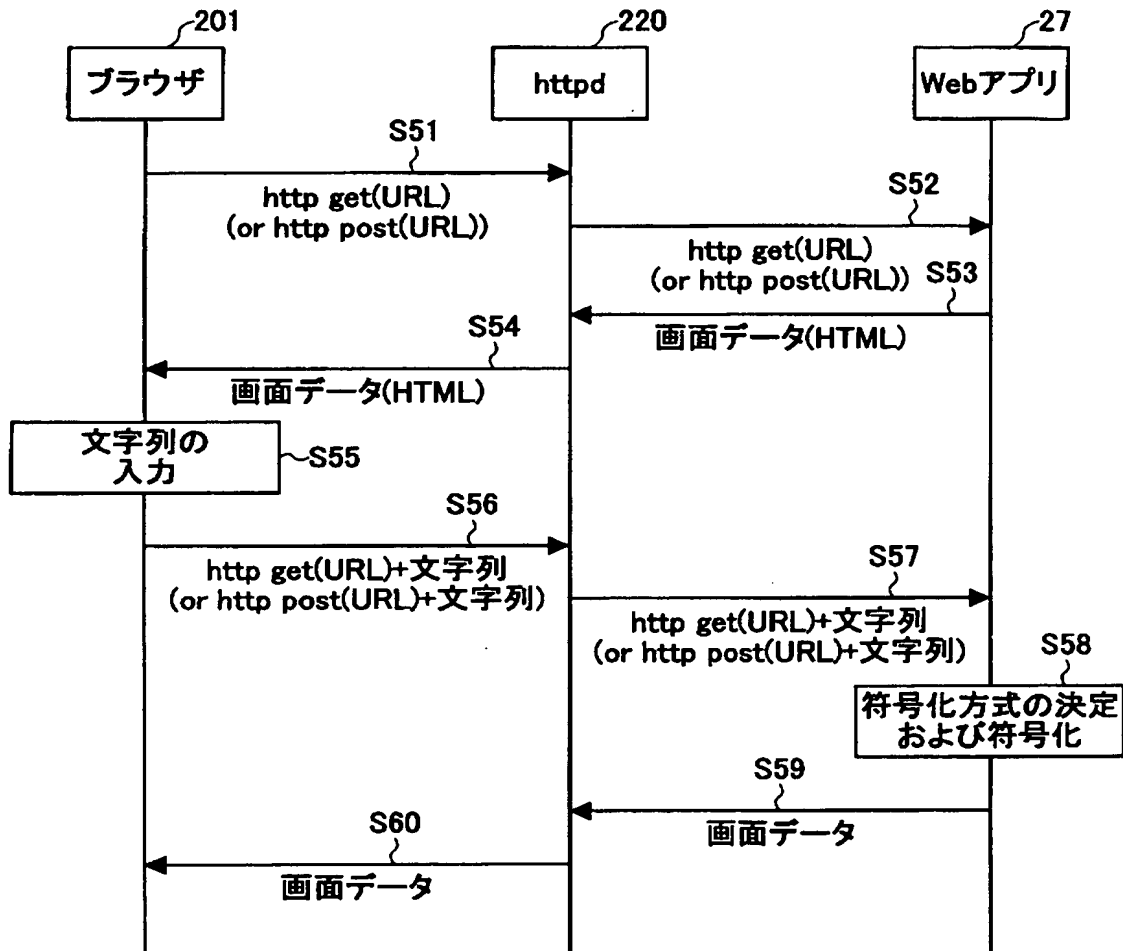


【図 22】

本発明による第3実施例の文字列処理方法
について説明するための融合機の構成図



【図 23】

本発明による第3実施例の文字列処理方法
について説明するためのシーケンス図

【書類名】 要約書**【要約】**

【課題】 一の符号化方式で符号化された文字列を他の符号化方式に変換するときに、適切な符号化方式を選択することができ、変換後の符号化方式で表現された文字列を適切に扱うことが可能な文字列処理装置、文字列処理方法および画像形成装置を提供することを目的とする。

【解決手段】 一の符号化方式で符号化された文字列を、複数の符号化方式から選択された他の符号化方式に変換する文字列処理装置 1 において、複数の符号化方式を選択し、選択した符号化方式を用いて前記一の符号化方式で符号化された文字列を変換したときの代用符号の数情報および位置情報の少なくとも一方を複数の符号化方式ごとに取得して、数情報および位置情報の少なくとも一方に基づき他の符号化方式を決定する符号化方式決定手段 2 3 0 を有することにより上記課題を解決する。

【選択図】 図 1 0

特願 2 0 0 4 - 0 0 4 1 2 3

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 6 7 4 7]

1. 変更年月日	2 0 0 2 年 5 月 1 7 日
[変更理由]	住所変更
住 所	東京都大田区中馬込 1 丁目 3 番 6 号
氏 名	株式会社リコー